

Faculdade de Engenharia da Universidade do Porto

Simulação de um Sistema Automático de Logística Interna para a Indústria de Calçado

Paulo Jorge da Rocha e Silva Sá Marques

Licenciado em Engenharia Electrotécnica e de Computadores pela Faculdade de
Engenharia da Universidade do Porto

Dissertação submetida para satisfação parcial dos requisitos do grau de mestre em
Automação, Instrumentação e Controlo

Dissertação realizada sob supervisão de
Professor Doutor António Ernesto da Silva Carvalho Brito,
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto do
Departamento de Engenharia Mecânica e Gestão Industrial

Porto, 31 Julho de 2007

Resumo

Um dos principais problemas que a indústria de calçado tem vindo a sentir é que os mercados cada vez mais procuram lotes de menores dimensões e com prazos de entrega mais curtos. As indústrias anteriormente estavam mais ajustadas a lotes de grandes dimensões e por isso, as metodologias que funcionavam eficientemente deixaram de o fazer. Há necessidade de preparar as indústrias para responder melhor a estes caprichos do mercado. Essa preparação passa entre várias coisas por uma melhor organização interna, como por exemplo, tornar as implantações ‘layout’ mais racionais. A distribuição física dos equipamentos pode não estar adequada às quantidades e à diversidade de produtos produzidos. Poderá haver custos constantes de movimentação de materiais e pessoas que poderiam ser minimizados. É então de vital importância a optimização dos processos e em particular a redução de custos, tempos de produção e transporte entre as várias secções de produção nas empresas industriais.

A empresa de calçado Codizo resolveu minimizar os custos de toda a movimentação de contentores com produtos em curso de fabrico na planta fabril, implantando um sistema de movimentação e armazenamento completamente automático que permitisse então o transporte do produto em curso em contentores (caixas), entre as secções de produção.

O sistema desenvolvido e instalado na empresa é composto por um sistema automático de movimentação e armazenamento intermédio do produto em curso de fabrico e componentes, de uma forma agregada para cada uma das secções da empresa. Este sistema de armazenamento está dividido em quatro armazéns. É um sistema automático de transporte composto por uma vasta rede de tapetes e torres elevatórias que efectua a interface entre os armazéns e as secções de fabrico.

A empresa tem necessidade de ter em simultâneo um conjunto alargado de gamas operatórias diferentes. O sistema de movimentação e armazenamento tem desempenhos de funcionamento dependentes de um conjunto de variáveis.

O tema central desta dissertação é a criação de um modelo de simulação (utilizando a ferramenta de simulação Arena) que possibilite testar o desempenho do sistema de movimentação e armazenamento instalado na empresa, através de um conjunto de cenários criados pelo utilizador.

O modelo de simulação desenvolvido é apresentado através da utilização da animação, permitindo mostrar a sequência realizada pelos contentores, assim como o armazenamento dos mesmos.

Abstract

Currently one of the shoe industry's biggest issues is that today's market seeks the fulfilment of product orders for smaller production lots with shorter deadlines. This industry's previous production methodologies were specifically adapted to large production lots. As a result of the current market trend such methodologies are now obsolete. A need therefore arises to respond to these market changes. Such a response includes amongst other things improving the internal organization of the enterprise by for example rationalizing the production floor layout. The physical distribution of equipment and production cells may also not be the most adequate for the quantity and diversity of the products that are produced. In addition to this, fixed costs associated with the improvement of material and people may be minimized. Process optimization in the factories is therefore of vital importance specifically in regards to cutting costs, reducing production time and transportation time that occur between the various production centres.

Subsequently the Codizo shoe factory decided to cut costs related to the transportation of all work in progress within the factory plant by implementing a fully automated storage and retrieval system that will allow for the movement of container between production cells.

The system that was developed and installed in the factory consists of two subsystems, an automated transportation system and an automated warehouse. Together they move and store both work in progress and material in an aggregated fashion so that these can be used in the various shop-floor sections. The automated warehouse itself has four storage and retrieval facilities. The automated transportation system on the other hand consists of a vast array of conveyors and elevation towers that make up the interface between the warehouses and production units.

The company needs to simultaneously manage a large set of differing production routes. The performance exhibited by the automated storage and retrieval system depends on a number of variables that parameterize these production routes.

The central theme of this dissertation is the creation of a simulation model using the Arena simulation software tool. It allows one to analyse the performance of the automate storage and retrieval system based on a number of scenarios created by a user.

The simulation model that is developed is presented through the use of animation. This allows one to visualize the sequences of movements and storage and retrieval operations that a container undergoes.

Prefácio

Esta dissertação vem na sequência de um sistema de logística interna instalado numa empresa de calçado. O sistema foi desenvolvido pelo INESC Porto em parceria com a empresa Lirel. Foi um dos sistemas desenvolvidos no âmbito do projecto FATEC, com início em Julho de 2002 e conclusão em Dezembro de 2005. Tratou-se de um projecto mobilizador para o desenvolvimento tecnológico, enquadrou-se no programa PRIME, que foi promovido pelo Centro Tecnológico do Calçado. O INESC Porto desenvolveu o sistema de gestão operacional e uma parte da automação, a Lirel desenvolveu a outra parte da automação, a estrutura mecânica e a instalação eléctrica.

O objectivo da dissertação é o desenvolvimento de um modelo de simulação, utilizando a ferramenta de simulação Arena, que possibilite testar o desempenho do sistema de movimentação e armazenamento instalado na empresa, através de um conjunto de cenários criados pelo utilizador.

Agradeço ao meu orientador, Professor António Ernesto da Silva Carvalho Brito, pelo seu apoio e incentivo. A sua disponibilidade, críticas e sugestões, e sua forma amigável como me acompanhou ao longo do desenvolvimento foram imprescindíveis.

Agradeço também à minha esposa Adelaide Marques e à restante família (meus filhos Ivo e Eva) pela compreensão e paciência que têm tido comigo. O seu carinho e apoio têm sido fundamentais na minha decisão para voltar aos estudos e completar o mestrado.

A todos os amigos e colegas da Unidade de Engenharia de Sistemas de Produção do INESC Porto, que de alguma forma me incentivaram e ajudaram, o meu muito obrigado.

Índice

1	INTRODUÇÃO.....	1
1.1	Motivação.....	1
1.2	Estrutura da dissertação.....	1
2	SIMULAÇÃO DIGITAL	3
2.1	Introdução.....	3
2.2	Modelação de sistemas.....	4
2.3	Vantagens e desvantagens da simulação	6
2.3.1	Vantagens	6
2.3.2	Desvantagens.....	7
2.3.3	Causas de insucesso no desenvolvimento da simulação	7
2.4	O processo de simulação	8
2.5	O futuro da simulação.....	10
2.6	Classificação dos simuladores	11
2.7	Linguagens genéricas e de simulação	13
2.8	Ferramentas de simulação - simuladores	14
2.9	Linguagens de programação versus ferramentas de simulação.....	16
2.10	A ferramenta de simulação Arena	18
2.10.1	Histórico do software Arena	18
2.10.2	O software de simulação Arena	19
2.10.2.1	O ambiente de simulação Arena	25
2.10.2.2	A utilização do VBA no Arena:.....	29
2.10.2.3	Módulos do Arena:	30
3	CASO DE ESTUDO	32
3.1	Introdução.....	32
3.2	A empresa	33
3.2.1	Caixas	33
3.2.2	Secções	34
3.2.3	Gamas operatórias	36
3.2.4	Sistema de gestão operacional.....	38
3.2.5	Sistema de automação.....	39
3.2.6	Constituição do sistema de movimentação e armazenamento	40
3.3	Modo de funcionamento	45
3.4	Pedido de caixas.....	49
4	DESENVOLVIMENTO DO MODELO	55
4.1	Introdução.....	55
4.2	Desenvolvimento do caso de estudo	55
4.2.1	Lógica de interface entre os armazéns e a logística	56
4.2.2	Arranque do sistema	58
4.2.2.1	O módulo ‘Create’:.....	58
4.2.2.2	O módulo ‘Assign’:.....	59
4.2.2.3	O bloco ‘VBA’:	60
4.2.3	Criar a entidade ‘caixa’	61
4.2.4	Configuração do sistema.....	65
4.2.5	Reentrada de caixas no sistema	68
4.2.5.1	O módulo ‘Hold’:.....	68
4.2.5.2	O módulo ‘Process’:	70
4.2.6	Lógica das torres.....	72
4.2.6.1	O módulo ‘Access’:	72
4.2.6.2	O módulo ‘Convey’:	73
4.2.6.3	O módulo ‘Station’:	74
4.2.6.4	O módulo ‘Exit’:	75
	O bloco VBA na entrada das secções:.....	76
4.2.6.5	O módulo ‘Decide’:	77

O bloco VBA na saída das secções:	79
4.2.7 Lógica dos tapetes.....	81
4.2.7.1 Ligações entre tapetes - Logística	82
4.2.7.2 Ligações entre tapetes - Armazéns.....	87
Lógica de entrada das caixas nos armazéns por gravidade.....	88
Lógica de saída das caixas nos armazéns por gravidade	90
Lógica de entrada das caixas no armazém num plano horizontal.....	92
Lógica de saída das caixas no armazém num plano horizontal	94
4.2.8 Animação.....	96
4.2.8.1 O módulo ‘Conveyor’	96
4.2.8.2 O módulo ‘Segment’	97
4.2.8.3 ‘Layout’	98
5 RESULTADOS	101
5.1 Introdução.....	101
5.2 Avaliação dinâmica do sistema em análise.....	101
5.3 Verificação e validação do modelo.....	101
5.4 Experimentação e análise de resultados	105
5.4.1 Análise de sensibilidade	107
5.4.1.1 Seleccionar o conjunto de variáveis de entrada e parâmetros do sistema que maior impacto causam nas variáveis de saída	107
5.4.1.2 Seleccionar as variáveis de saída de maior importância	108
5.4.1.3 Executar o modelo para as combinações dos valores das variáveis de entrada e parâmetros do sistema	109
5.4.1.4 Quantificar o impacto causado nas variáveis de saída.	111
5.4.2 Comparação de cenários	113
5.4.2.1 Escolha dos cenários a comparar	113
5.4.2.2 Configuração dos modelos para cada um dos cenários.....	114
5.4.2.3 Comparação dos resultados nos vários cenários	114
6 CONCLUSÕES E DESENVOLVIMENTOS FUTUROS.....	117
6.1 Conclusões gerais	117
6.2 Perspectivas de desenvolvimento futuro	118
7 REFERÊNCIAS E BIBLIOGRAFIA	120
A EXEMPLO DE CÓDIGO	126

Índice de Figuras

Figura 2-1: Tipos de simulação (Berends (1999))	4
Figura 2-2: Relação entre Processos, actividades e eventos	11
Figura 2-3: Esforço com "software" de desenvolvimento	15
Figura 2-4: Complexidade da simulação por computador	16
Figura 2-5: Número de 'papers' submetidos na conferencia WSC 2006	18
Figura 2-6: Estrutura hierárquica do 'software' Arena (Adaptado de Kelton (2002))	21
Figura 2-7: Arquitectura dos produtos Arena	22
Figura 2-8: Arena Basic de "ACT Solutions"	23
Figura 2-9: Arena Profissional de "ACT Solutions"	23
Figura 2-10: Arena Enterprise de "ACT Solutions"	24
Figura 2-11: Arena Factory Analyser de "ACT Solutions"	24
Figura 2-12: Arena Contact Center de "ACT Solutions"	25
Figura 2-13: Módulos da 'template' Basic Process	31
Figura 3-1: Contentor utilizado para o transporte de produto interno	34
Figura 3-2: Secções no Sistema de armazenamento e movimentação instalado	36
Figura 3-3: Construção da gama operatória através de tabela	37
Figura 3-4: Construção da gama operatória graficamente	37
Figura 3-5: Arquitectura cliente-servidor	38
Figura 3-6: Locais de entrada/saída de caixas	39
Figura 3-7: Interface PC - PLC	39
Figura 3-8: Tapetes utilizados.....	42
Figura 3-9: Torre elevatória	42
Figura 3-10: Localização dos armazéns da empresa.....	43
Figura 3-11: Exemplo dos armazéns 101,102 e 103.....	43
Figura 3-12: Sentido de deslocamento dos Caixas nos armazéns.....	44
Figura 3-13: Armazém num plano horizontal (armazém 104)	44
Figura 3-14: Sentido de movimentação das caixas no armazém 104.	45
Figura 3-15: Local das torres	46
Figura 3-16: Montagen 2 Enformar (Torre 3).....	47
Figura 3-17: Configuração das listas numa secção.....	47
Figura 3-18: Área de memória no PLC onde são escritas as listas de caixas	48
Figura 3-19: Exemplo de gamas operatórias	49
Figura 3-20: Exemplo de gamas operatória	49
Figura 3-21: Sequência de pedidos de caixas das torres aos armazéns	50
Figura 3-22: Fluxograma de uma torre a pedir uma caixa para uma secção	51
Figura 3-23: Lista de caixas pedidas pelas torres	52
Figura 3-24: Entrada de caixa numa secção	53
Figura 3-25: Saída de uma caixa de uma secção	54
Figura 4-1: Sistema de Simulação	56
Figura 4-2: Arranque do sistema.....	58
Figura 4-3: Criação das entidades do sistema.....	59
Figura 4-4: Criação dos atributos das caixas	60
Figura 4-5: Chegada de caixas utilizando em ficheiro Excel	61
Figura 4-6: Armazéns que abastecem as secções	62
Figura 4-7: 'Query' à Base de dados instalada na Codizo.....	62
Figura 4-8: Configuração de cada armazém	66
Figura 4-9: Configuração de cada secção	66
Figura 4-10: Configuração das Torres	67

Figura 4-11: Écran de arranque do sistema.....	68
Figura 4-12: Entrada no sistema	70
Figura 4-13: Processo Subcontratados (Proveniente da criação caixas).....	70
Figura 4-14: Processo de SubContratados proveniente do arranque	71
Figura 4-15: Lógica da Torre 8	72
Figura 4-16: Lógica da T 1	72
Figura 4-17: Módulo Access.....	73
Figura 4-18: Módulo 'Convey'	73
Figura 4-19: Módulo 'Convey' Dados	74
Figura 4-20: Modulo 'Station'	75
Figura 4-21: Módulo 'Exit'	75
Figura 4-22: Módulo VBA entrada das Secções.....	76
Figura 4-23: 'Buffer' entrada nas secções	77
Figura 4-24: Módulo 'Decide'	78
Figura 4-25: Módulo 'process' nas torres	78
Figura 4-26: Módulos saída das torres	80
Figura 4-27: Módulo Access saída de uma torre	80
Figura 4-28: Módulo Convey na saída de uma torre	81
Figura 4-29: Ligação tipo 1E1S	82
Figura 4-30: Ligação tipo 2E1S	82
Figura 4-31: Simulação c/ ligação tipo 2E1S	83
Figura 4-32: Módulo 'decide' na ligação 2E1S	83
Figura 4-33: Ligação do tipo 1E2S	84
Figura 4-34: Simulação c/ ligação tipo 1E2S hipótese 1	84
Figura 4-35: Módulo 'decide' na ligação 1E2S – hipótese 1	85
Figura 4-36: Simulação c/ ligação tipo 1E2S - hipótese 2.....	85
Figura 4-37: Módulo 'decide' na ligação 1E2S – hipótese 2.....	86
Figura 4-38: Lgação nE1S	86
Figura 4-39: Ligação tipo nE2S.....	86
Figura 4-40: Ligação tipo entroncamento.....	87
Figura 4-41: Entrada/Saída Armazém	88
Figura 4-42: Lógica entrada armazém gravidade	89
Figura 4-43: Armazém gravidade segura caixa	89
Figura 4-44: Módulo 'decide' na entrada do armazém de gravidade	89
Figura 4-45: Lógica de saída dos armazéns de gravidade	90
Figura 4-46: Módulo 'Hold' na lógica de saída dos armazens por gravidade	90
Figura 4-47: Fim da lógica saída dos armazéns de gravidade	91
Figura 4-48: Lógica na entrada armazém horizontal	92
Figura 4-49: Módulo 'Decide' entrada armazém horizontal.....	93
Figura 4-50: Lógica final entrada armazém horizontal.....	94
Figura 4-51: Módulo 'Hold' no armazém horizontal.....	94
Figura 4-52: Lógica de saída caixas dos alvéolos no armazem horizontal.....	94
Figura 4-53: Lógica da saída de caixas no armazém horizontal.....	95
Figura 4-54: Módulo de dados 'Conveyor'.....	96
Figura 4-55: Módulo 'Segment'.....	97
Figura 4-56: 'Layout' das estações	98
Figura 4-57: 'Layout' dos segmentos	99
Figura 4-58: 'Layout' Final.....	100
Figura 5-1: Fases de um estudo de simulação (adaptado de Law e McComas (1991))	102

Figura 5-2: Ecrã de arranque e paragem do modelo	103
Figura 5-3: Exemplo de taxa de utilização na secção 8	116

Índice de tabelas

Tabela 2-1: Exemplo de simuladores disponíveis	15
Tabela 4-1: Gama 1	64
Tabela 4-2: Gama 2	64
Tabela 4-3: Gama 3	64
Tabela 4-4: Gama 4	64
Tabela 4-5: Gama 5	64
Tabela 4-6: Gama 6	64
Tabela 4-7: Gama 7	65
Tabela 4-8: Secções - Torres	65
Tabela 4-9: Distribuições no Arena (Prado(2004))	79
Tabela 5-1: Quantidade de pares pretendidos em cada secção	109
Tabela 5-2: Parâmetros da simulação	110
Tabela 5-3: Gamas operatórias no sistema tipo G2	110
Tabela 5-4: Taxa de utilização dos recursos homem nas secções c/ uma replicação	112
Tabela 5-5: Número de caixas realizadas em cada secção c/ uma replicação	112
Tabela 5-6: Taxa média de utilização dos recursos homem nas secções c/ cinco replicações	112
Tabela 5-7: Média de caixas realizadas em cada secção c/ cinco replicações	113
Tabela 5-8: Taxa média de utilização dos recursos homem nas secções c/ dez replicações	113
Tabela 5-9: Média de caixas realizadas em cada secção c/ dez replicações	113
Tabela 5-10: Gamas operatórias no sistema tipo G1	114
Tabela 5-11: Gama tipo 1, com 10 pares por caixa	114
Tabela 5-12: Gama tipo 1 e com nº pares por caixa a variar	114
Tabela 5-13: Gama tipo 1 e com demasiados pedidos de caixas	115
Tabela 5-14: Gama do tipo 2 e com pedidos de caixas previstos	115
Tabela 5-15: Gama tipo 2 e com o dobro de pedidos previstos	115

Abreviaturas

<u>Termo</u>	<u>Definição</u>
---------------------	-------------------------

AST	- Aplication Solution Template
CAD	- Computer Aided Design
DAO	- Data Access Objects
CLS	- Control and Simulation Language
ERP	- Enterprise Resource Planning
FATEC	- Fabrica de Alta Tecnologia para Calçado
FIFO	- First In, First Out
GPSS	- General Purpose Simulation System
GPSS/H	- General Purpose Simulation System H (Hardcover)
IDE	- Integrated Development Environment
MFC	- Microsoft Fundation classes
OOS	- Object Oriented Simulation
ORMS	- Magazine for members of the Institute for Operations Research and the Management Sciences
PLC	- Programmable Logic Controllers
PRIME	- Programa de Incentivos à Modernização da Economia
SIMAN	- SIMmulation ANalysis language
SLAM	- Simulation Language for Alternative Modeling
SQL	- Structured Query Language
VBA	- Visual Basic for Applications

1 Introdução

1.1 Motivação

Actualmente, a maioria das empresas industriais tem que competir em mercados que se caracterizam pela procura de produtos de elevada qualidade, com um considerável nível de personalização, tecnologicamente inovadores e com os menores custos e prazos de entrega possíveis. Como consequência, as encomendas dos clientes caracterizam-se por serem cada vez mais frequentes, e conterem uma crescente variedade de produtos. As empresas industriais, para se manterem competitivas, têm então que ajustar os seus sistemas de produção a esta nova realidade.

Esta dissertação considera como alvo de estudo o sistema de logística interna que permite automatizar a movimentação e o armazenamento de matéria-prima e produtos em curso de fabrico numa planta fabril. Este sistema é constituído por armazéns automáticos, tapetes transportadores e torres elevatórias. Os produtos em curso de fabrico das ordens de produção lançadas em produção são movimentadas neste sistema em contentores com dimensões em média de 30cm x 60cm x 50cm. Na generalidade dos casos os tapetes transportadores e os armazéns automáticos são colocados a um nível elevado (3 metros do chão), sendo a sua deslocação na vertical efectuada por torres elevatórias. O sistema desenvolvido utiliza vários PLC's (Programmable Logic Controller), têm uma componente de automação e de gestão complexa e realiza as seguintes funções: gestão das gamas operatórias dos contentores e a gestão do transporte dos contentores entre secções de produção e entre secções e armazéns, utilizando tapetes com diversos cruzamentos.

Este sistema permite melhorar a organização, o controlo de todo o processo produtivo, a automação de tarefas não produtivas, tais como movimentação de produto em curso e matérias-primas, que trazem vantagens a vários níveis.

1.2 Estrutura da dissertação

A dissertação encontra-se dividida em cinco capítulos principais.

No **capítulo 2** efectua-se uma abordagem à modelação dos sistemas, as vantagens e desvantagens da utilização da simulação. São apresentados alguns casos de insucesso no desenvolvimento da simulação. É realizada uma apresentação do processo de simulação, é apresentada uma perspectiva do que poderá ser a simulação no futuro. A classificação dos simuladores existentes, e uma apresentação das linguagens e

ambientes de simulação. É ainda apresentado o dilema da escolha de uma linguagem ou de uma ferramenta de simulação para o desenvolvimento de um projecto e o porquê de ter sido escolhido a ferramenta de simulação Arena. Por último é apresentada uma descrição do Arena.

O **capítulo 3** descreve todo o sistema de movimentação e armazenamento instalado na empresa Codizo e que será objecto deste estudo. Inicia-se pela apresentação dos contentores utilizados para o transporte dos produtos em curso na empresa, de seguida é realizada uma breve descrição das secções de produção na empresa. Posteriormente é apresentado o modo de construção das gamas operatórias entre secções, a arquitectura do sistema de gestão operacional do sistema instalado e o sistema de automação responsável pela movimentação e armazenamento dos contentores. Neste capítulo é apresentada também a constituição de todo o sistema de movimentação e armazenamento, assim como o seu modo de funcionamento. Por fim é apresentado o algoritmo utilizado pelo PLC responsável pela movimentação e pelos pedidos de contentores (caixas) ao sistema de armazenagem e que se pretende que o modelo de simulação represente.

No **capítulo 4** descreve-se o desenvolvimento do modelo de simulação, começando por apresentar em mais detalhe a ferramenta de simulação Arena. De seguida é apresentada a implementação do algoritmo para a realização dos pedidos de caixas aos armazéns de produto em curso. Todo o processo de arranque, desde a criação das entidades, à criação das gamas operatórias, à reutilização das entidades no sistema, até aos ficheiros de configuração, é apresentado também neste capítulo. Por último é apresentada, a lógica utilizada na construção do modelo, nos pontos de maior interesse, como os armazéns, tapetes com cruzamentos e torres elevatórias.

No **capítulo 5** é apresentada uma avaliação dinâmica do sistema em análise, a verificação e validação do modelo. No âmbito da experimentação e análise dos resultados é feita a análise de sensibilidade e uma comparação de cenários no sistema de simulação, recorrendo à ferramenta “Process Analyser” do Arena.

O **capítulo 6** apresenta as principais conclusões desta dissertação. Após a apresentação das conclusões mais importantes são dadas algumas indicações sobre a possível evolução do trabalho desenvolvido, nomeadamente no sentido de melhorar o desenvolvimento de um modelo para os sistemas de movimentação e armazenamento de produto em curso.

2 Simulação Digital

2.1 Introdução

Este capítulo apresenta os principais conceitos da Simulação Digital, salientando a sua aplicação para a modelação e avaliação de desempenho de sistemas discretos.

Brito e Feliz (2001), referem que “Um sistema é todo e qualquer objecto sobre o qual se pretende realizar um determinado estudo, enquanto um modelo é uma qualquer representação desse objecto na qual se irá, efectivamente, executar tal estudo”. É importante referir que os resultados através de um modelo virão sempre afectados de incertezas uma vez que o modelo será sempre uma “aproximação” ao sistema real e não o verdadeiro sistema (Brito e Feliz (2001)).

Segundo Freitas Filho (2001), um sistema é um conjunto de elementos interligados que actuam para atingirem um objectivo. Os elementos podem ser diversas coisas, tais como, matérias-primas, pessoas, objectos etc. que executam tarefas e constituem processos, enquanto que um modelo é uma representação de um sistema que requer alguma abstracção e simplificação do funcionamento do sistema real.

Simular é desenvolver um modelo de um sistema em projecto ou já construído com o objectivo de avaliar o seu comportamento segundo vários aspectos, sendo possível desta forma tirar conclusões sem necessitar de o construir (Kelton et al. (2002)).

Ainda Kelton, et al. (2002), segundo um ponto de vista prático, a simulação é o processo de concepção e criação de um modelo computadorizado de um sistema proposto ou real com o propósito de conduzir experimentos numéricos para nos dar um melhor entendimento do comportamento do sistema através de determinadas condições.

Modelar um sistema é em muitos casos uma vantagem no desenvolvimento de projectos, porque permite antever o comportamento e desempenho dos sistemas sem custos avultados num tempo relativamente curto. Nos sistemas já em funcionamento permite prever o seu desempenho nos casos extremos sem prejudicar as suas actividades.

Quando falamos de simulação podemos ter vários tipos de modelos. Segundo Berends (1999), este distingue-os conforme Figura 2-1

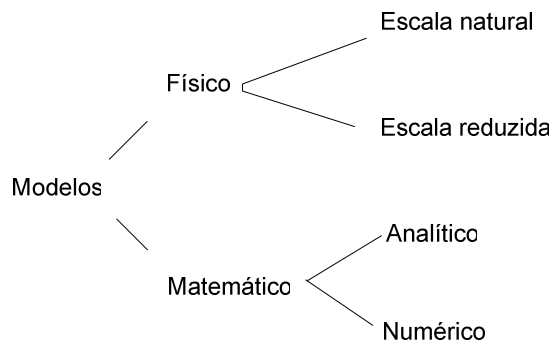


Figura 2-1: Tipos de simulação (Berends (1999))

Os modelos podem então ser classificados de duas maneiras: físico ou matemático. Nos modelos matemáticos, podemos ainda distinguir entre analíticos ou numéricos. Nos analíticos, a sua representação é feita através de equações matemáticas, enquanto, que nos modelos numéricos a sua representação é feita através da simulação digital.

Segundo Kelton, et al. (2002), se um modelo é suficientemente simples será mais vantajoso a utilização de ferramentas matemáticas tais como a teoria das filas, métodos de equações diferenciais, ou mesmo por vezes programação linear. Contudo a maior parte dos sistemas estudados são bastante complicados e por isso obter modelos válidos é também bastante complicado. É nestes casos que a simulação deve ser utilizada.

De uma maneira geral os métodos analíticos poderão resultar em soluções mais económicas e mais precisas, contudo nos casos de modelos complexos, torna-se inviável a sua utilização. Um exemplo deste tipo de modelos analíticos, são os modelos de programação linear. Para os casos mais complexos utilizam-se os modelos numéricos, nestes casos as equações que relacionam as variáveis do sistema não são conhecidas, e será este o caso do tipo de simulação considerada nesta dissertação.

2.2 Modelação de sistemas

Existem várias maneiras para classificar os modelos de simulação, Kelton et al. (2002) refere que os modelos de simulação podem ser classificados segundo três perspetivas:

- Estático versus Dinâmico
- Contínuo versus Discreto
- Determinístico versus Estocástico

Estático versus Dinâmico: nos modelos estáticos, o tempo não tem um papel natural enquanto que nos modelos dinâmicos tem. Segundo Almeida (1999), sempre que as mudanças de estado não envolvam tempo, estamos em presença de modelos estáticos. Segundo Kelton e al. (2002), a maior parte dos modelos operacionais são modelos

dinâmicos, por exemplo, nos casos em que envolvem sistemas de produção, onde o tempo decorrido para as mudanças de estados é um factor relevante.

Determinístico versus Estocástico: modelos de simulação que não contêm variáveis aleatórias são classificados como determinísticos. Um exemplo será uma operação agendada com serviços temporais fixos (Kelton, e al. (2002)). Modelos estocásticos por outro lado contêm variáveis aleatórias. Sempre que num modelo se entra em conta com flutuação dos valores de certas variáveis, o processo de simulação deixa de ser considerado determinístico para tomar a designação de estocástico (Brito e Feliz (2001)), ou seja, têm uma ou mais variáveis aleatórias como entradas que levam a saídas aleatórias, por outras palavras, uma entrada válida pode causar várias mudanças no estado do modelo, devido aos processos aleatórios. Para introduzir no modelo um comportamento estocástico que determinadas variáveis do sistema exibem há que considerar dois casos distintos (Brito e Feliz (2001)): (1) ou tal comportamento é descrito através de uma distribuição de valores retirados de experiências levadas a cabo sobre o sistema, o que muitas vezes é dado em forma de tabela de frequências de valores ou de histogramas; (2) ou tal comportamento é descrito de forma analítica, seja através de funções adaptadas ao conjunto dos dados experimentais, seja usando funções de probabilidades típicas, como a distribuição Normal, a distribuição de Poisson, ou outras.

Contínuo versus Discreto: modelo de simulação contínuo é aquele em que as variáveis dependentes podem variar continuamente ao longo do tempo simulado, o estado do sistema pode variar continuamente ao longo do tempo. Um exemplo é um reservatório de água, o nível da água poderá subir ou descer dependendo da precipitação ou da evaporação. Num modelo discreto pode haver mudanças mas em separados espaços temporais, as variáveis dependentes variam em espaços temporais específicos do tempo da simulação. Segundo Kelton et al. (2002), ainda poderemos ter ambos os casos, contínuo e discreto no mesmo modelo, chamados nestes casos modelos mistos contínuos-discretos. Na maior parte dos modelos estamos em presença de modelos mistos, contudo predomina uma das partes.

Os modelos de simulação podem ainda correr em duas modalidades de tempo:

1- Tempo, real, em que a escala de tempo é a real, os eventos ocorrem e são tratados numa mesma escala de tempo que corresponde ao sistema real. Os Simuladores de jogos ou de treino enquadram-se nesta categoria. Nestes sistemas um operador humano interage com o simulador em tempo real.

2- Tempo simulado, neste caso, não acompanha a escala de evolução do tempo real. Um ano do tempo de simulação pode decorrer em poucos segundos de processamento. São utilizados para análises de desempenho em que o interesse é pelas medidas de desempenho.

Na construção de um modelo, o analista não precisa de incluir todos os detalhes do sistema. O objectivo do estudo é que define a quantidade de detalhes a incorporar no modelo. O analista deverá definir a estrutura do modelo a construir, identificando quais as entidades envolvidas, os seus atributos e as actividades do sistema. Devem

também ser definidos os atributos e as interações entre as entidades devem estar definidas.

2.3 Vantagens e desvantagens da simulação

Nesta secção são abordadas as vantagens, desvantagens e causas de insucesso, (Law e Kelton, (1991), (Paiva (2005)), e texto de vários autores, compilados por Bressan (2002)).

2.3.1 Vantagens

A simulação apresenta um conjunto de vantagens:

- A simulação pode ser utilizada em sistemas complexos com elementos estocásticos que não conseguem ser descritos perfeitamente com os modelos matemáticos;
- No instante do desenvolvimento dos modelos dos sistemas reais obriga os responsáveis a entenderem o papel de cada componente do sistema e quais as possíveis interações entre eles;
- O desenvolvimento do modelo de simulação ajuda os responsáveis a separar as variáveis e os parâmetros controláveis daqueles que não são controláveis e a estudar a influência de cada um deles sobre os sistemas;
- As conclusões resultantes da simulação permitem que os responsáveis avaliem os recursos a utilizar, ou ainda como os recursos disponíveis poderão ser alocados;
- A simulação, de uma maneira geral, aplica-se à análise de problemas de grande complexidade que não podem ser resolvidos por técnicas tradicionais com modelos analíticos;
- A possibilidade de utilização de várias distribuições de probabilidade, conforme as particularidades do problema real;
- Permite a análise de sensibilidade do tipo “o que aconteceria se...”. Vários cenários podem ser testados e comparados rapidamente, ou seja, permite a replicação precisa das experiências, podendo-se, assim, testar alternativas diferentes para o sistema;
- Permite uma análise individual de cada componente ou variável do modelo para determinar quais são os mais importantes;
- Com a possibilidade de realizar um conjunto de replicações no modelo alterando as variáveis de entrada que mais afectam os resultados, permite um melhor controlo sobre as condições experimentais;
- Permite simular longos períodos, num tempo reduzido;
- A simulação permite a identificação de problemas resultantes de especificações mal concebidas e com “bottlenecks” evitando em muitas ocasiões a sua construção ou modificação não necessitando desta maneira de investimentos avultados e condenados ao fracasso.

Em conclusão, poderemos dizer que a simulação é uma boa ferramenta útil no apoio à tomada de decisão, permitindo uma melhor avaliação da solução em estudo.

2.3.2 Desvantagens

A simulação apresenta também algumas desvantagens:

- Um modelo de um sistema complexo pode ter um custo elevado e levar vários meses para ser desenvolvido, especialmente nos casos em que os dados são de difícil obtenção e não coerentes; A simulação não gera resultados fiáveis sem “inputs” adequados. A construção e a alimentação do modelo requerem um trabalho árduo e criterioso;
- Os resultados da simulação, quando apresentados em grande número, com efeitos de animações e gráficos, podem levar a uma confiança nos resultados acima da justificada. Se o modelo não for uma representação válida do modelo em estudo, este não terá utilidade, mesmo que os resultados causem boa impressão. A simulação é muito dependente da validade do modelo desenvolvido, ou seja, de nada adianta fazer-se um estudo detalhado dos dados de saída e encontrar-se uma solução para o problema, se o modelo criado não representa fidedignamente o sistema ou se os dados de entrada não são correctos;
- Cada execução da simulação estocástica produz apenas estimativas dos parâmetros analisados. Apesar dos novos “software” de simulação possuírem ferramentas poderosas que podem ajudar na obtenção de bons resultados, a simulação não é uma técnica optimizante, ela só testa as alternativas dadas pelo utilizador;
- Os modelos não são reutilizáveis em outros sistemas. Cada modelo de simulação é único. A não reutilização de um modelo em diferentes situações, pode prejudicar a possibilidade de ganhos de tempo consideráveis.

2.3.3 Causas de insucesso no desenvolvimento da simulação

Segundo Brito e Feliz (2001), o desenvolvimento do modelo é uma tarefa que poderá ser bastante demorada, é aconselhável que quem desenvolve possua suficiente destreza para definir o grau de precisão a utilizar, não se perdendo no desenvolvimento de um modelo muito complexo, quando os mesmos resultados se poderiam obter através de um modelo mais simples.

Seguidamente são apresentadas algumas das causas de insucesso no desenvolvimento da simulação, que são referidas com frequência por vários autores:

- No arranque do desenvolvimento da simulação não ter os objectivos bem definidos;

- Nível inadequado de detalhes: Pouco detalhe ou muito detalhe;
- A falta de comunicação entre quem desenvolve e os responsáveis pelo sistema simulado durante o desenvolvimento da simulação;
- Uma má interpretação do sistema a desenvolver pela equipa que realiza o desenvolvimento do modelo;
- Falha de compreensão dos resultados da simulação por parte dos responsáveis.
- Tratar a simulação de forma amadora, como um simples exercício;
- Falha em formar uma equipa com conhecimentos de metodologias e técnicas de simulação;
- Falha na obtenção de dados representativos do comportamento do sistema;
- Software de simulação inadequado;
- Ferramenta de simulação muito complexa e com documentação inadequada;
- Crença de que ferramentas de simulação sofisticadas e com recursos amigáveis, não necessitam de conhecimentos técnicos da teoria de simulação;
- Utilização de animação não adequada;
- Falha na não consideração ou na consideração inadequada dos factores aleatórios no comportamento do sistema em estudo;
- A utilização de distribuições incorrectas, isto é, que não correspondem ao comportamento real, como dados de entrada da simulação;
- Análise dos dados de uma execução da simulação utilizando fórmulas que supõem independência;
- Executar uma única vez a simulação e considerar os dados obtidos como a resposta verdadeira (tempo de execução inadequado);
- Utilizar medidas de desempenho inadequadas.

2.4 O processo de simulação

O objectivo principal da simulação é ajudar os responsáveis pelo sistema a tomarem uma decisão sobre a resolução do problema apresentado. Quem desenvolve os modelos de simulação deve ser conhecedor das boas técnicas de soluções de problemas de simulação e ao mesmo tempo deve ser conhecedor das boas práticas a utilizar no desenvolvimento dos modelos de simulação (Pedgen (1995)). No desenvolvimento de projectos de simulação, os seguintes aspectos devem estar considerados (Bank (1996)):

Formulação do problema: Nesta fase são definidos os objectivos da simulação. Os responsáveis pelo sistema de simulação conjuntamente com os responsáveis pelo sistema real devem especificar os resultados que esperam obter da simulação, os vários cenários a realizar na simulação, as variáveis que interessa analisar, o tratamento a dar aos resultados, caso seja necessário, qual a melhor interface a apresentar ao utilizador.

Construção do modelo: Esta é uma etapa fundamental, porque é aqui que é caracterizado o problema, e por isso deve ser realizada com muito cuidado. São especificados os factores que influenciam o sistema. Deve haver uma relação de compromisso entre ter um modelo de fácil leitura e que ao mesmo tempo traduza e reflecta as características mais importantes do sistema real. Os aspectos mais

relevantes do sistema dependem então de como o problema é formulado. Inicialmente deverá começar por ser um modelo simples que será enriquecido à medida que vai sofrendo um conjunto de iterações. Cada informação recebida ou recolhida deverá ser analisada ao nível de objectivos ou restrições.

Definição dos dados de entrada: Os dados de entrada dependem da formulação do problema e do modelo. Os seus valores podem ser provenientes de dados históricos, baseados numa análise preliminar, experiência ou medições realizadas no sistema real. A apresentação dos dados deve ter uma formatação adequada a ser utilizada posteriormente no modelo.

Implementação do modelo: Nesta fase, o modelo é implementado através de código de computador. Consiste na tradução do modelo anteriormente definido numa forma que o computador entenda. A tradução poderá ser mais complexa ou mais simples, conforme a ferramenta de programação ou simulação utilizada. A velocidade de implementação também depende da ferramenta computacional utilizada e da experiência do analista. Existe no mercado, um conjunto alargado de ferramentas de simulação que facilitam a tradução do modelo.

Verificação: Nesta fase o analista verifica utilizando um conjunto de procedimentos, se o modelo traduzido no computador corresponde ao que estava previsto na construção do modelo. Aqui são realizados os testes necessários até se obter a confirmação de que o modelo está bem traduzido. Com a variação dos valores de entrada será verificada a coerência dos resultados. Ferramentas de simulação com animação são um forte contributo para uma mais rápida verificação do modelo.

Validação: Esta etapa consiste em verificar se o modelo executado representa adequadamente o sistema real. Numerosos testes são realizados sobre o modelo para avaliar a sua qualidade. Se o resultado dos testes se aproximar dos valores apresentados no sistema real, dentro de um nível de confiança aceitável pelos responsáveis e pelos analistas, a simulação será validada. Esta confrontação deverá ser realizada junto dos responsáveis do sistema, porque são eles que conhecem bem o sistema real. É nesta etapa que se verifica se o modelo desenvolvido está consistente e é transmitida a confiança aos responsáveis do sistema real.

Plano de experimentação: Esta fase refere-se ao estabelecimento do plano de experiência a realizar e a estabelecer as condições experimentais para a execução da simulação. Consiste em executar o modelo desenvolvido utilizando diversos cenários de simulação estabelecidos. Aqui decide-se quando começa a simulação, as variáveis a realçar nos cenários, o número de simulações e qual o tempo de execução da simulação. Se o analista utilizar ferramentas apropriadas poderá realizar uma quantidade de testes consideráveis e verificar de uma maneira rápida qual o cenário mais apropriado.

Execução do modelo: Com base no plano de experimentação o modelo é executado. Aqui a execução deve ter como objectivo principal a diminuição das variações nas amostras. É a fase da execução da simulação para gerar os resultados pretendidos e realizar a análise de sensibilidade.

Análise dos resultados: Esta fase consiste em determinar se os resultados obtidos com a simulação correspondem aos resultados esperados. Os resultados retirados da simulação são um conjunto de valores, que necessitam muitas das vezes de serem tratados e analisados. As actividades que maior impacto causam, e as mais críticas são avaliadas nesta fase. É ainda uma fase difícil e que poderá ser longa.

Implementação e documentação: Durante esta fase o analista pode afinar o modelo fazendo pequenos ajustes, deve ainda dar formação ao utilizador e garantir a validação dos resultados. Nesta fase é realizada a execução do projecto no sistema real. O registo dos procedimentos realizados ao longo de todo o projecto deve ser realizado para que na documentação exista uma evolução do historial das várias fases de desenvolvimento. A documentação é sempre importante durante todo o desenvolvimento, para futuras alterações ou mesmo para que futuramente o modificador possa melhorar os seus métodos de desenvolvimento.

Todas as etapas descritas anteriormente são dinâmicas, isto é, são contínuas as melhorias ao longo de todo o desenvolvimento do modelo, deve haver um refinamento em cada um das etapas. As mudanças devem ser adaptadas com a particularidade do sistema sobre o qual está a ser desenvolvido o modelo.

2.5 O futuro da simulação

De acordo com J. Carson e D. Brunner (2000) haverá no futuro um aumento significativo na utilização da simulação. Esta será embebida noutras aplicações de software, e a simulação será usada em mais larga escala em decisões de tempo real em vez dos métodos tradicionais. A literatura geral sugere que a interligação do software de simulação com outro software é crucial. Os formatos dos dados dos software de simulação usados nos modelos e previsões dos comportamentos dos sistemas de produção e das aplicações de projectos, engenharia de produção, e a necessidade de gestão da produção serem os mesmos. As especificações de uma interface neutra que permitiriam a integração rápida e fácil do software comercial devem ser desenvolvidas.

Taylor e Robinson (2006) referem que num projecto de simulação, cerca de 30-70 % do tempo é dispendido na recolha de dados. No sentido de melhorar este processo de recolha vários processos podem ser melhorados, tais como, o processo de recolha ser mais automatizado, principalmente nos casos dos modelos que são repetidamente usados, garantir que os dados recolhidos são exactos; os dados serem adequados e actuais, e o que fazer no caso dos dados recolhidos estarem incompletos e não disponíveis.

Segundo Yücel (2005), uma outra previsão importante acerca do futuro da simulação é o desenvolvimento de interfaces standard que ajudaria ao desenvolvimento da tecnologia utilizada na simulação. Actualmente em muitos casos, o processo de desenvolvimento, é trabalho intensivo, talvez mais uma arte do que uma ciência, deixa um trabalho e uma responsabilidade considerável ao analista da simulação. Uma das

ideias promotoras para expandir a simulação a um leque mais amplo de utilizadores é o conceito de haver já alguns modelos pré-construídos ou componentes de modelos que possam ser introduzidos no modelo a ser desenvolvido. A ideia é seleccionar os componentes de uma biblioteca e usá-los directamente. O objectivo é construir uma única vez, um componente, e torná-lo disponível numa biblioteca de maneira a que possa ser utilizável em várias aplicações. Existem já algumas ferramentas que permitem a construção de ‘templates’ que podem ser utilizadas para automatizar processos, contudo muito há ainda que desenvolver neste campo, porque a sua construção só pode ser realizadas por especialistas.

2.6 Classificação dos simuladores

A base para a classificação dos modelos resulta de como as variáveis de estado se alteram ao longo do tempo. O tempo é a variável independente. As restantes variáveis incluídas na simulação são funções do tempo e, portanto, são variáveis dependentes.

Os simuladores classificam-se em discretos ou contínuos, ou seja, de acordo com os tipos de modelos que trabalham, e como as variáveis dependentes mudam o seu estado ao longo do tempo. Se as variáveis dependentes variam discretamente em instantes específicos do tempo, estamos em presença de modelos de mudança discreta (simulação discreta). Se as variáveis dependentes variam continuamente ao longo do tempo, estamos em presença de modelos de mudança contínua (simulação contínua).

Um modelo deve descrever como um sistema executa um determinado trabalho ou processo. As linguagens de simulação discreta, para descreverem o comportamento dinâmico dos sistemas discretos utilizam três componentes (processos, actividades e eventos ou acontecimentos). Segundo Soares (1992) um sistema é representado de uma forma dinâmica como um conjunto de processos interactivos, e cada um desses processos é composto por diversas actividades, cujas interacções são controladas por eventos. A relação entre processos, actividades e eventos pode ser visto na Figura 2-2.

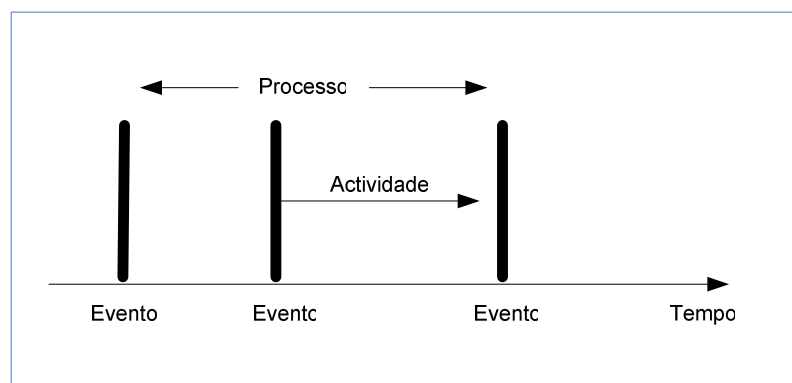


Figura 2-2: Relação entre Processos, actividades e eventos

Na modelação de sistemas discretos, os simuladores podem ainda ser classificados, como orientados ao processo, à actividade, ao evento e às três fases.

Um simulador é **orientado a processos** se o programa for organizado como um conjunto de processos executados concorrentemente durante a simulação. Se a sequência de processos é executada sequencialmente e se cada processo gere um conjunto de eventos do mesmo tipo diz-se que estamos em presença de um simulador orientado a processos (Wagner (2000)). Contudo a sequência com que os eventos são lançados não é necessariamente a mesma do sistema real. Em vez de se representar o fluir da entidade por actividades ou eventos, é estabelecido o percurso completo dela desde que entra até que abandona o sistema. O sistema passa, assim, por ser representado segundo o ponto de vista das entidades activas que nele intervêm, e o que interessa é descrever o percurso que cada uma dessas entidades deverá realizar no seu ciclo de vida (Brito e Feliz (2001)).

Basicamente uma simulação orientada a processo deve seguir os seguintes passos (SPOL(1994)):

- definir as entidades do sistema
- criar um processo para cada objecto do sistema descrevendo as suas etapas
- executar concorrentemente os processos.

Segundo Brito e Feliz (2001) a abordagem por processo é a forma mais simples de descrever a dinâmica de alguns sistemas. A maior desvantagem desta abordagem resulta da dificuldade de interligar processos que pertencem a entidades diferentes.

Numa simulação **orientada às actividades**, o analista descreve o conjunto das actividades detectadas no sistema, especificando com o máximo pormenor a sequência de acções a realizar em cada uma dessas actividades (Brito e Feliz (2001)). Ainda segundo (Brito e Feliz (2001)), todas as actividades são analisadas em todos os incrementos da simulação, mesmo as actividades que não sejam executadas nesse espaço temporal, apesar do programa ter informação suficiente para saber que em determinados instantes algumas actividades não necessitavam de ser analisadas. Por ser uma abordagem pouco eficiente, é uma simulação pouco utilizada na modelação de sistemas ((Soares (1992)). Contudo apesar da pouca eficiência ainda por vezes é utilizada nas fases de arranque de alguns projectos de simulação porque permite uma implementação simples e rápida das actividades (Brito e Feliz (2001)).

Um simulador **orientado a eventos**, a dinâmica do sistema é representada segundo um encadeamento de eventos, dando particular atenção do analista nos instantes da simulação nas transições de estado previstas no sistema (Brito e Feliz (2001)). Criar um modelo segundo esta abordagem equivale, por isso, a descrever, para cada evento relevante do sistema, o conjunto de acções “instantâneas” a ele associadas, acções essas que deverão ser executadas sempre que esse evento surja na simulação. Os estados do sistema são alterados em função dos eventos fazendo com que o estudo destes seja essencial à compreensão do comportamento de um sistema. Esta abordagem da simulação é bastante popular devido à sua eficiência e à sua versatilidade. Aqui o analista ao criar o modelo não se preocupa com a ordem do código a ser escrito, porque cada rotina de evento é executada de forma independente das outras (Brito e Feliz (2001)). Esta abordagem permite ainda conhecer o estado de qualquer entidade do sistema em qualquer instante de tempo. Contudo, esta

abordagem, por utilizar uma lista de eventos, os processadores utilizados consomem grande parte do seu tempo, na reorganização da referida lista (Brito e Feliz (2001)).

A abordagem **das três fases** introduzida por Tocher na década de 60, combina por um lado a simplicidade evidenciada na abordagem orientada às actividades, e por outro lado a eficiência evidenciada na abordagem por eventos, e ao mesmo tempo pretende ignorar ou minimizar o que de menos bom tem cada uma delas. Neste caso não existe lista de eventos em cada ciclo de eventos, e por esse facto esta abordagem torna-se mais rápida que a abordagem por eventos (Brito e Feliz (2001)).

2.7 Linguagens genéricas e de simulação

Nos primeiros desenvolvimentos de simulação foram utilizadas linguagens de programação genéricas como FORTRAN nos anos 60 (Kelton et al. (2002)). Estas simulações exigiam um enorme esforço no seu desenvolvimento, o que tornava muitas em muitos casos inviável o uso da simulação.

Na década de 60, com o aparecimento das linguagens orientadas à simulação, tornou-se um pouco mais fácil modelar sistemas. Posteriormente foram desenvolvidas outras ferramentas que adicionadas às linguagens de simulação, resultaram em ambientes de simulação mais poderosos para projectar sistemas.

As linguagens de programação utilizadas na implementação de um modelo podem ser de três classes básicas:

- Linguagens de programação genéricas ou convencionais;
- Extensão funcional;
- Linguagens específicas de simulação;
- Ferramentas ou ambientes de uso específico de simulação.

As linguagens de **programação genérica**, requerem que o analista seja bom programador e ao mesmo tempo um especialista em simulação para poder criar um ambiente de simulação. Por outro lado quem desenvolve não necessita de conhecer várias linguagens de programação, utiliza a linguagem que melhor conhece e desta forma poderá recorrer-se de todos os recursos que a linguagem pode utilizar. Exemplos típicos de linguagens genéricas utilizadas na simulação são: C, C++, FORTRAN e Java.

As linguagens referidas anteriormente aliadas de bibliotecas dedicadas tais como Simlib desenvolvida em C ou JavaSim desenvolvida para Java tornam-se mais poderosas e bastante flexíveis e permitem a implementação de um modelo de uma forma mais eficiente. Nestes casos estamos em presença de **extensões funcionais**.

Pelo facto de haver um intenso uso de simulação como uma forma de abordagem de análise de desempenho de sistemas deu origem ao aparecimento de um conjunto de linguagens **específicas de simulação** que são desenhadas para a modelação de sistemas de vários tipos (Soares (1992)). Estas linguagens já têm uma interface entre o

programador e a linguagem de simulação. São linguagens que têm todas as estruturas necessárias aos ambientes de simulação, um exemplo de linguagem específica de simulação, é a linguagem SIMAN.

Com estas linguagens o utilizador tem disponível, um conjunto de facilidades para a transformação do modelo conceptual do sistema num programa. Estas linguagens têm disponíveis funções e rotinas destinadas a análises estatísticas e controle do avanço do tempo na simulação.

Com as linguagens específicas houve uma evolução bastante significativa, e durante algum tempo deram resposta às solicitações dos analistas. Contudo com o aparecimento de sistemas cada vez mais complexos as exigências também aumentaram. Era necessário que além de trazerem resultados fiáveis, as linguagens de simulação deveriam mostrar os resultados de uma forma mais imediata e ao mesmo tempo mais amigável. Surgiram então as animações, que são funcionalidades integradas nos simuladores, com a facilidade de mostrarem aos utilizadores os sistemas graficamente. Ficou mais fácil ao cliente final da simulação perceber de uma forma quase directa os problemas evidenciados das simulações. Mesmo para o próprio analista tornou-se muito mais fácil a detecção de erros no desenvolvimento do modelo e na verificação e validação do mesmo. A própria formação aos utilizadores finais, tornou-se mais fácil. Um exemplo de “softwares” que foram pioneiros na utilização da animação, foi a linguagem CINEMA.

Conforme referido na subsecção anterior (2.7), as linguagens são classificadas: orientadas a processo, actividade ou evento, e a escolha da linguagem a utilizar dependerá se o modelo é orientado a processo, actividade ou evento. Como exemplo de linguagens orientadas a processo SIMULA (Brito e Feliz (2001)) e SIMAN (Say (1995)). Exemplo de linguagem orientada a actividade CLS (Rodrigues (1996)). Como exemplos de linguagens orientadas a evento, Slam II (Banks (2000)) e SIMSCRIPT (Macdougall (1975)).

2.8 Ferramentas de simulação - simuladores

Nas duas abordagens referidas anteriormente existem vantagens e desvantagens. Na primeira, com a utilização de linguagens de programação, temos maior flexibilidade, mas mais tempo e esforço no desenvolvimento, mesmo tratando-se de linguagens de simulação, com características próprias para o desenvolvimento de modelos de simulação. Apesar destas últimas linguagens já incorporarem animações, mesmo assim, eram também dispendiosas, como referido anteriormente pelo facto do desenvolvimento dos programas só poder ser feito eficazmente de uma maneira geral por especialistas, ou informáticos. O mesmo acontecia quando era necessário acrescentar ou alterar um determinado programa já desenvolvido. Com a utilização de ferramentas dedicadas estes inconvenientes poderão ser evitados.

Com as ferramentas de simulação (simuladores) o analista tem à sua disposição e em ambiente integrado, a possibilidade de descrever o modelo, controlar a simulação e visualização de estatísticas, com recursos para visualização, animação e tratamento de

dados. Uma das vantagens destas ferramentas é a sua aprendizagem ser mais fácil assim como o seu manuseamento, contudo é na mesma necessária alguma formação na ferramenta.

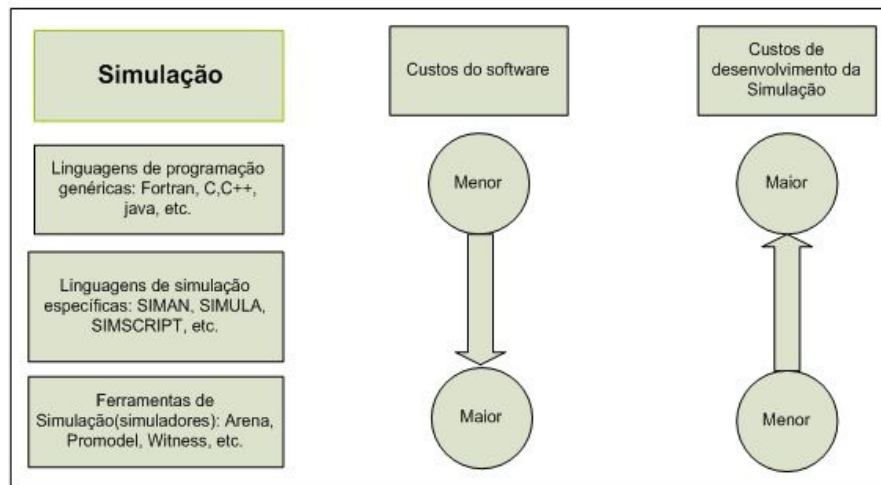


Figura 2-3: Esforço com "software" de desenvolvimento

A Figura 2-3 exemplifica o que foi referido anteriormente, e observa-se a relação entre o custo do software utilizado e o custo do desenvolvimento de um modelo de simulação.

Ambientes de simulação de alto nível apresentam interfaces amigáveis e permitem a criação e a animação de modelos de sistemas de forma bastante flexível. Exemplos de ambientes de simulação, alguns de domínio público estão referidos na Tabela 2-1. Muitas outras linguagens ou softwares poderão encontrar-se no mercado.

Ferramentas de Simulação	Endereços Web
Automod Simul8	http://www.autosim.com http://www.simul8.com
Promodel	http://www.promodel.com
Arena	http://www.arenasimulation.com
Witness	http://www.lanner.com/
Flexsim	http://www.flexsim.com/
Extend	http://www.imaginetthatinc.com/
GoldSim	http://www.goldsim.com/
Mast	http://www.cmsres.com/
SimCad	http://www.createasoft.com/

Tabela 2-1: Exemplo de simuladores disponíveis

2.9 Linguagens de programação versus ferramentas de simulação

No desenvolvimento de um sistema deparamo-nos com a escolha do “software” de simulação mais apropriada a utilizar. Na escolha são considerados factores tais como:

- Disponibilidade orçamental para a compra da ferramenta;
- “know how” da equipa de desenvolvimento responsável pelo desenvolvimento da simulação;
- Características do sistema a ser simulado;
- Características dos “softwares” disponíveis;
- Horizonte temporal disponível para a realização da simulação.

A utilização de uma ferramenta de simulação passa por várias opções conforme Figura 2-4 que confronta a flexibilidade dos “softwares” utilizados na simulação com a facilidade de utilização por parte do utilizador (Paiva (2005)). Desde a opção por pacotes de software de carácter mais genérico, com requisitos de programação e de tempos baixos, mas com uma flexibilidade limitada, até linguagens de programação com uma grande flexibilidade, mas com requisitos de programação e de tempos bastante elevados

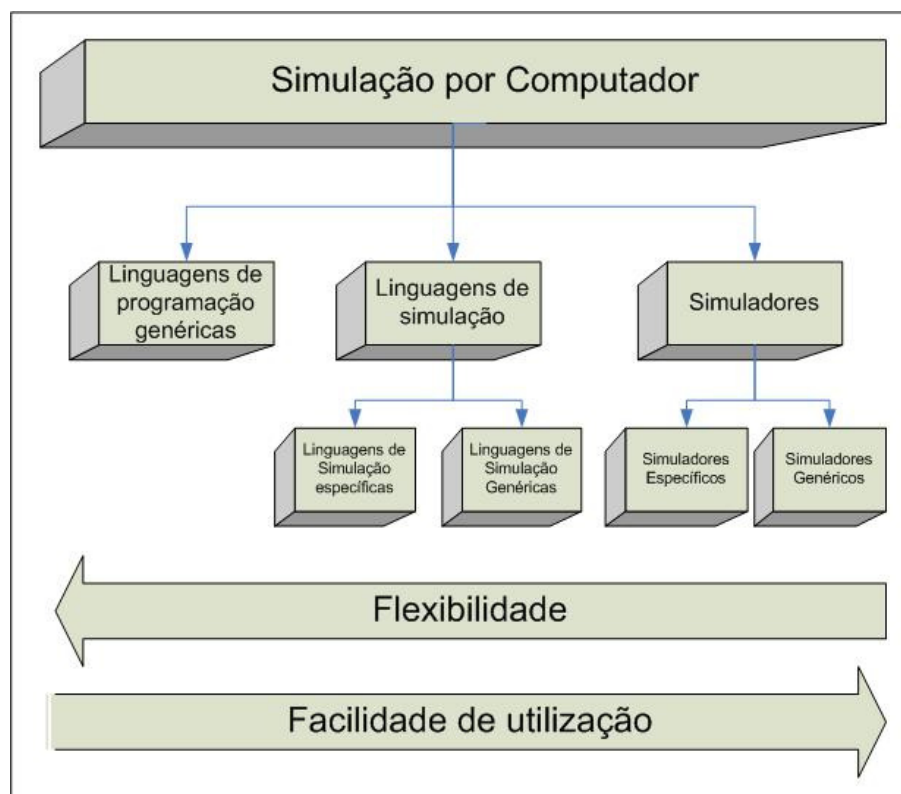


Figura 2-4: Complexidade da simulação por computador

Pelas razões atrás referidas não é de estranhar que cada vez mais se utilizem os pacotes de simulação em detrimento da utilização de linguagens de simulação. O sucesso dos pacotes de simulação fez com que aparecessem no mercado dezenas deles, o que poderá levar a quem está a iniciar-se na simulação a ficar com grandes dúvidas sobre qual o melhor ou mais apropriado.

A “ORMS Today” realiza dois “surveys” por ano aos pacotes de simulação mais utilizados. No “survey” de Dezembro de 2005 foram analisados 48 produtos que correm em computadores dando particular ênfase aos que executam simulações discretas. Produtos mais vocacionados para a simulação contínua ou treino foram omitidos. Neste “survey” foram considerados vários aspectos:

- Nome do software, vendedor, aplicações típicas, mercados primários de cada software;
- Requisitos do sistema:
 - Sistema operativo, RAM.
- Construção do modelo:
 - Construção gráfica, possibilidade de utilização de uma linguagem de programação, Debug em run-time.
- Distribuição de entrada, desenho experimental, suporte para análise de saída
 - Optimização, reutilização do código, partilha do modelo, ferramentas de suporte e custos pela ferramenta;
 - Análise de custos, misto discreto/contínuo.
- Animação
 - Vista em tempo real, exportar animação, compatibilidade, animação 3D, Importação de desenhos CAD.
- Suporte/Formação
 - Suporte/linha de suporte, grupo de discussão na área, Cursos de formação, formação, formação num site, consultadoria.
- Informação sobre preços
 - Standard, versões para estudantes.
- Melhores características desde 2001
- Comentários dos vendedores

Na combinação dos vários aspectos referidos no exame, o pacote ARENA sobressaiu pelas suas características. Foi o único a ter uma boa referência em todas as características. As suas excelentes características, aliadas a ser gratuito para estudantes faz com que cada vez mais se utilize em todo o mundo, no meio académico e empresarial.

O software Arena aparece referenciado em “Modular Design and Integration of Simulation tools”, outro exame a vários pacotes de simulação discreta, como um software com grandes facilidades de troca de dados e grande facilidade de integração com outras ferramentas.

Num estudo realizado pela Rockwell Software (Figura 2-5), mostrou que na conferência sobre simulação discreta ‘Winter Simulation Conference WSC 2006’, dos mais de 300 ‘papers’ apresentados, quarenta e oito por cento, quase metade, foram sobre simulações utilizando a aplicação Arena. Nesta comparação foram incluídos os seguintes produtos Arena, AutoMod, ProModel, Extend, Simul8, AnyLogic, Enterprise Dynamics, FlexSim, CSIM, Micro Saint, eM-Plant, SIMSCRIPT, Witness, e iGrafx.

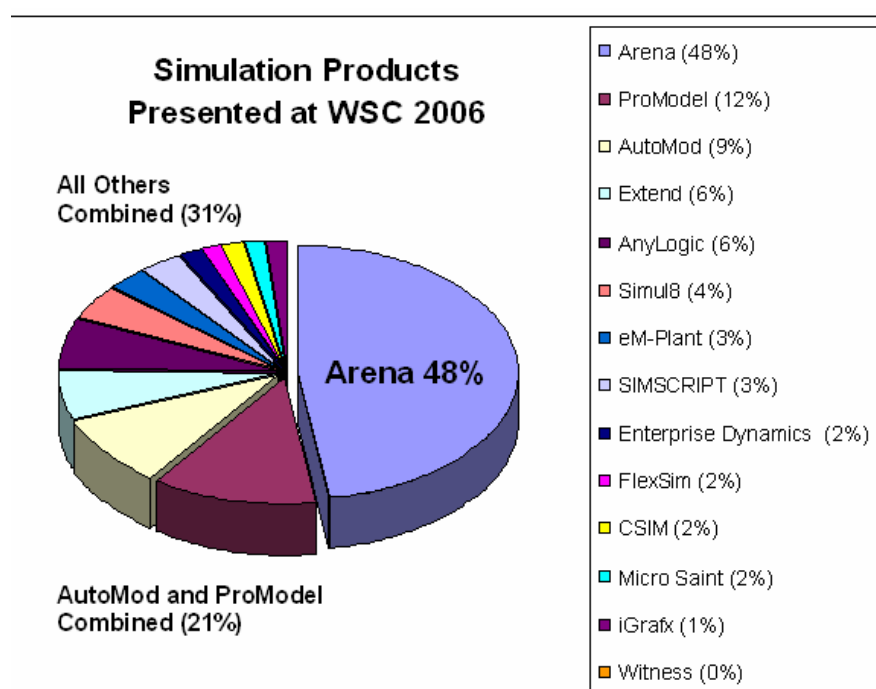


Figura 2-5: Número de ‘papers’ submetidos na conferencia WSC 2006

Também por ser um software genérico, com enorme potencial e por estar disponível no INESC Porto, foi a aplicação escolhida para a realização da simulação deste trabalho.

2.10 A ferramenta de simulação Arena

2.10.1 Histórico do software Arena

Em 1982 foi lançada a primeira versão da linguagem de simulação SIMAN pela “Systems Modelyng Corporation (EUA)”, inspirada na linguagem GPSS usada em

computadores de grande porte. Inovadora, foi a primeira linguagem específica de simulação destinada a IBM-PC compatíveis.

Nos anos seguintes, o SIMAN foi aprimorado e também a sua interface, inovando novamente: foi a primeira linguagem de simulação a fazer uso do rato na sua interface, numa época anterior ao Windows. Em 1990, foi lançado o pacote CINEMA, que, integrado ao SIMAN permitia, apresentar uma representação animada e em cores do funcionamento do sistema. Foi, mais uma vez, a primeira interface do tipo para simuladores.

Em 1993, SIMAN e CINEMA foram integrados num ambiente único de simulação que unia e potencializava seus recursos, o Arena. A linguagem SIMAN, através do Arena, passou a ser representada em formato gráfico, tornando-se bastante intuitiva. Em 1995 foi lançado o Arena para Windows 95, sendo a primeira ferramenta de simulação a trabalhar em versão 32 bits. No ano seguinte, com a versão 3.0, passou a ser a primeira a receber a certificação “Microsoft Windows Compatible”, integrando a linguagem VBA, que permite aceder ou ser acedido por todos os aplicativos do MS Office e outros.

Com a compra da “Systems Modeling” pela “Rockwell” em 2000, o ARENA recebeu um enorme impulso de desenvolvimento, e novas versões agregando melhorias são lançadas em intervalos de tempo cada vez mais curtos. O ARENA passou a fazer parte da suíte RS BIZWARE, que reúne uma solução integrada e completa para projectar, planear e gerir o chão de fábrica. Assim, o ARENA agora é parte importante da estratégia de actuação da “Rockwell Software”, braço de software da “Rockwell”, dentro do segmento de MES (“Manufacturing Execution System”).

2.10.2 O software de simulação Arena

O Arena une os recursos de uma linguagem de simulação, referidos anteriormente, à facilidade de uso de um simulador, num ambiente gráfico integrado, que contém todos os recursos para modelação de sistemas, desenho do modelo, animação das entidades, análise estatística e análise de resultados. O “software” tem como tecnologia de base incorporada a linguagem de simulação SIMAN e combina a facilidade de utilização dos simuladores de alto nível com a flexibilidade das linguagens de simulação e linguagens de programação genéricas (Kelton, et al. (2002)).

O “software” de simulação Arena constitui uma aplicação, compatível com o sistema Windows, que permite desenvolver modelos de simulação animados, para sistema discretos e contínuos. O Arena é ainda completamente compatível com outros “softwares” Windows, tais como o processador “Word”, folha de cálculo “Excel” e aplicações de software CAD.

Os modelos na aplicação Arena são construídos juntando blocos lógicos, construindo a lógica do modelo computadorizado representativo do modelo conceptual idealizado, através de um editor gráfico de alto nível. É possível construir modelos sem utilizar uma única linha de código do Arena, pelo facto da sua construção se basear na

utilização de um ambiente gráfico e visual de maneira integrada, não obrigando o programador a conhecer a linguagem de simulação SIMAN utilizada pelo Arena.

O Arena apresenta ao utilizador um conjunto de bibliotecas painéis que contêm diversos elementos gráficos de análise e modelação (módulos). A aplicação pode utilizar diversos tipos de painéis, inclusive simultaneamente. Com a aquisição do Arena “Standard” são fornecidos três painéis genéricos que podem ser utilizados para todos os tipos de processo (“Basic Process”, “Advanced Process” e “Advanced Transfer”). Todos juntos são designados por “Arena template”. Juntamente com a “template ” vem também dois painéis SIMAN (blocos e elementos) designados por “SIMAN template”

No Arena a ligação ao utilizador é o ícone. Este representa um bloco ou um elemento que executa uma função específica ou um grupo de funções. Os parâmetros necessários para introduzir nas funções, podem ser inseridos numa tabela ou numa caixa de diálogo que aparece através de duplo toque, utilizando o rato no respectivo ícone.

Um “template” do Arena consiste num painel ou um conjunto de painéis que os analistas utilizam no desenvolvimento de uma aplicação, sistema, ou classes de sistemas. Num “template panel” estão incluídos dois tipos de blocos (dados e lógicos). O Arena permite ao utilizador criar as suas próprias “templates”, transformando o simulador genérico num simulador específico para várias áreas de aplicação, para proveito próprio ou para ser utilizado por outros utilizadores.

O Arena emprega uma arquitectura hierárquica na modulação, isto é, módulos são definidos utilizando outros módulos. Este tipo de arquitectura oferece vantagens porque os módulos que representam subconjuntos ou conjuntos de processos podem ser desenvolvidos e verificados uma vez, e posteriormente poderão ser utilizados para desenvolver módulos de processos de mais alto nível hierárquico.

Os módulos base do Arena representam a linguagem de simulação SIMAN. Estes módulos já referidos anteriormente representam a “SIMAN template” composta por dois painéis: Blocos e Elementos. A estrutura hierárquica do Arena apresentada na Figura 2-6, permite seleccionar diferentes componentes de ‘software’ de acordo com o nível de modelação pretendido e possibilita a sua integração na mesma interface gráfica.

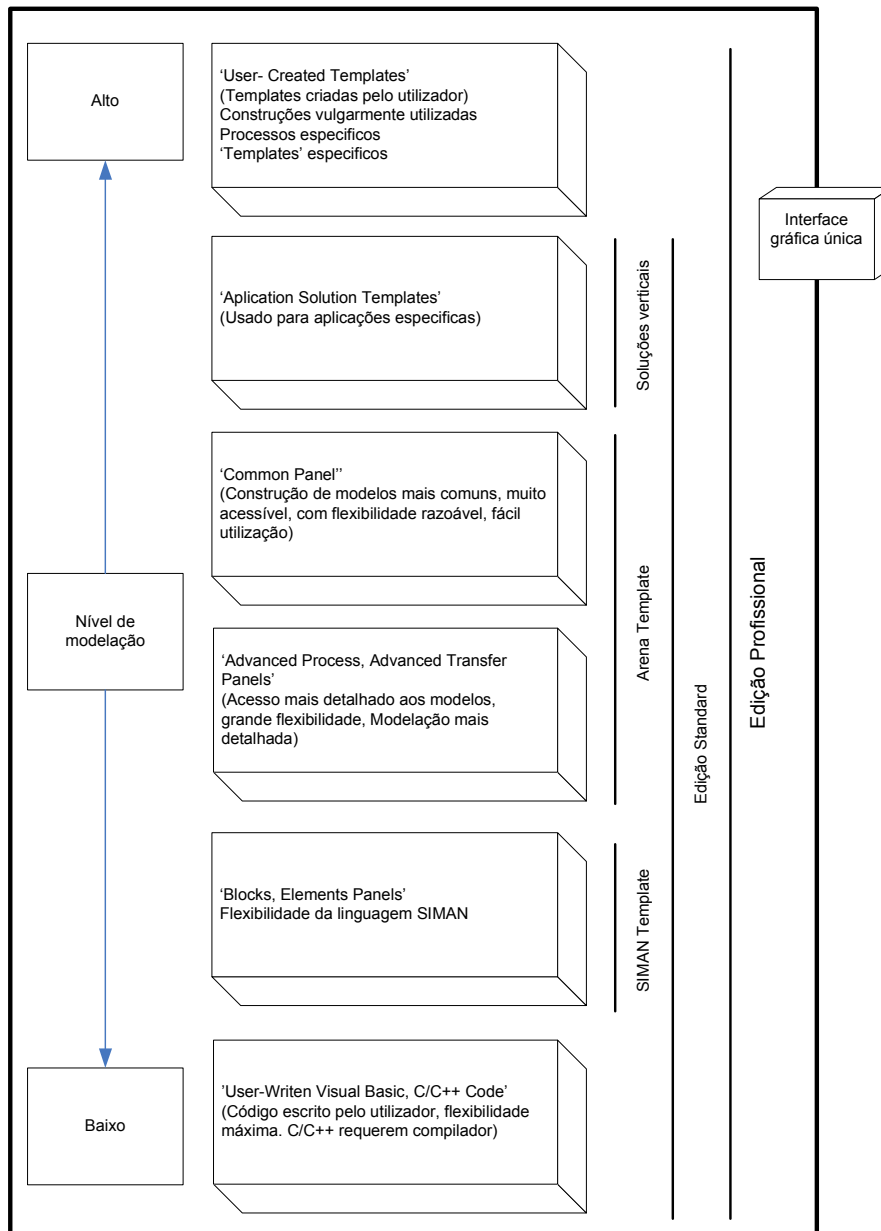


Figura 2-6: Estrutura hierárquica do 'software' Arena (Adaptado de Kelton (2002))

O 'software' Arena inclui vários produtos e várias ferramentas como opção, que se complementam e que permitem focar diferentes áreas de uma organização, apresentadas na Figura 2-7 retirado da Rockwell Automation.

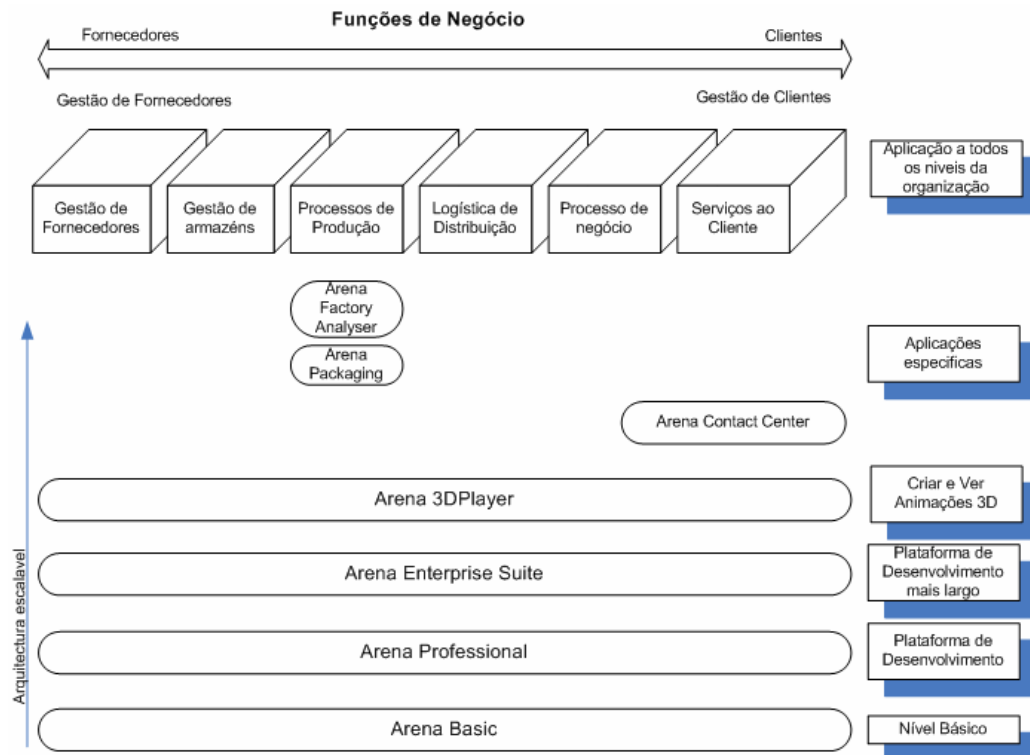


Figura 2-7: Arquitetura dos produtos Arena

A família Arena é composta então pelos cinco produtos: ‘Arena Basic’, ‘Arena Profissional’, ‘Arena Enterprise’, ‘Arena Factory Analyser’, e ‘Arena Contact Center’. Contém ainda como Opções que vêm acompanhar alguns dos produtos anteriormente referidos:

- ‘Packing’ (‘Template’);
- ‘Contact Center’ (‘Template’);
- ‘Arena 3DPlayer’ (‘To create and view 3D animations’);
- ‘OptQuest for Arena’ (‘optimization tool’);
- ‘Run Time’ (‘Simulation Player’)

Uma breve descrição dos produtos onde são referidas as ‘templates’ usadas e as ferramentas que acompanham os produtos retiradas da Rockwell:

- **Arena Basic:** para modelação e análise de processos de negócio, sistemas de produção, serviços e outros sistemas.

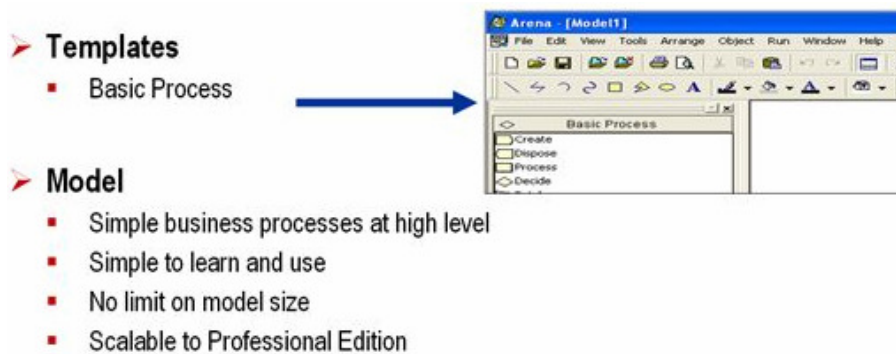


Figura 2-8: Arena Basic de “ACT Solutions”

Na ‘template’ ‘Basic Process’ estão os blocos básicos que são a estrutura principal na criação de um modelo. Sistemas mais simples, em que podem ser menosprezadas as distâncias entre processos, ou seja, os tempos em que não existam “valor adicionado” podem ser considerados instantâneos, são passíveis, de serem modelados simplesmente com os blocos básicos.

- **Arena Profissional:** para modelar, de um modo mais flexível e detalhado, diferentes tipos de sistemas; o ‘Arena template’ (Figura 2-7) inclui um conjunto de diferentes módulos que permitem modelar a maioria das aplicações e, especificamente, sistemas de produção e sistemas de gestão de materiais; esta versão inclui também ferramentas que apoiam o desenvolvimento de projectos de simulação: ‘Optquest for Arena’ que permite definir diferentes parâmetros de entrada (controladores e restrições) e medidas de desempenho desejadas (funções objectivo), testando vários cenários para alcançar as saídas desejadas; o programa combina, num único algoritmo de pesquisa, as metas-heurísticas de pesquisa Tabu, redes neurais e ‘scatter search’. Possibilidade de criação de bibliotecas (‘templates’ ou ‘Aplication Solution Templates – AST’s) específicas para determinadas aplicações alvo; os módulos criados em cada biblioteca representam componentes dos sistemas reais e encapsulam dados e lógica para reutilização nos modelos Arena

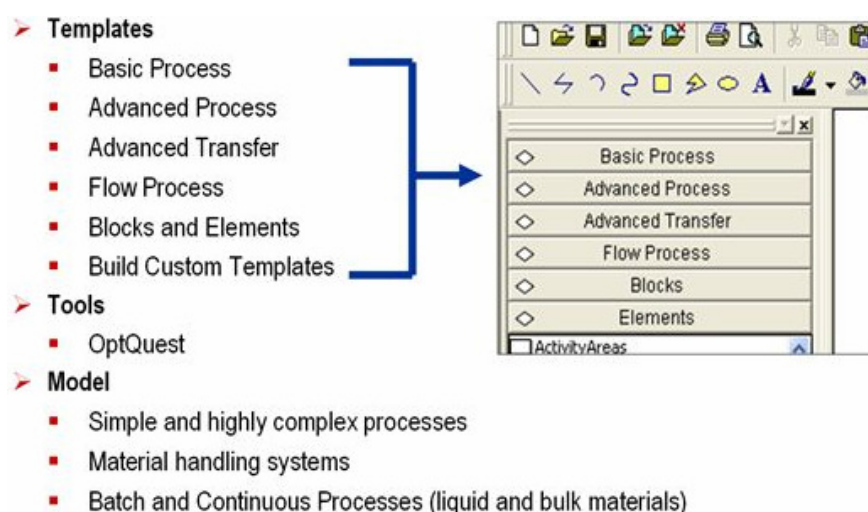


Figura 2-9: Arena Profissional de “ACT Solutions”

- **Arena Enterprise:** oferece uma conveniente quantidade de produtos que permite modelar uma larga quantidade de problemas. Inclui as características de ‘Arena Factory’, mais ‘Arena Contact Center’ e ‘Arena 3DPlayer’

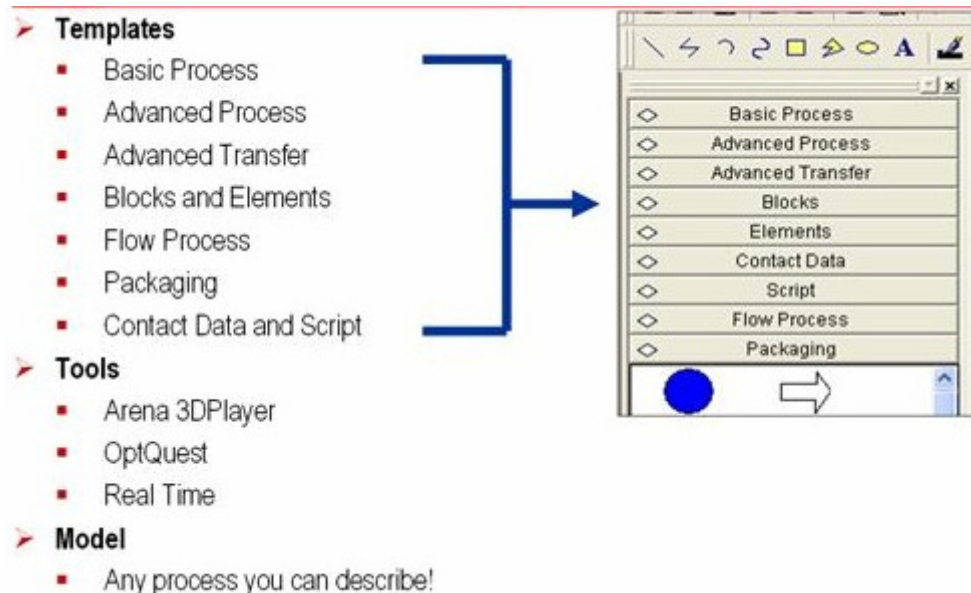


Figura 2-10: Arena Enterprise de “ACT Solutions”

- **Arena Factory Analyser:** é mais vocacionado para analisar linhas de produção com grandes velocidades e (produções realizadas continuamente), que é fundamental para os processos de produção de várias indústrias, incluindo alimentação, farmacêutica, electrónica, química, saúde e beleza.

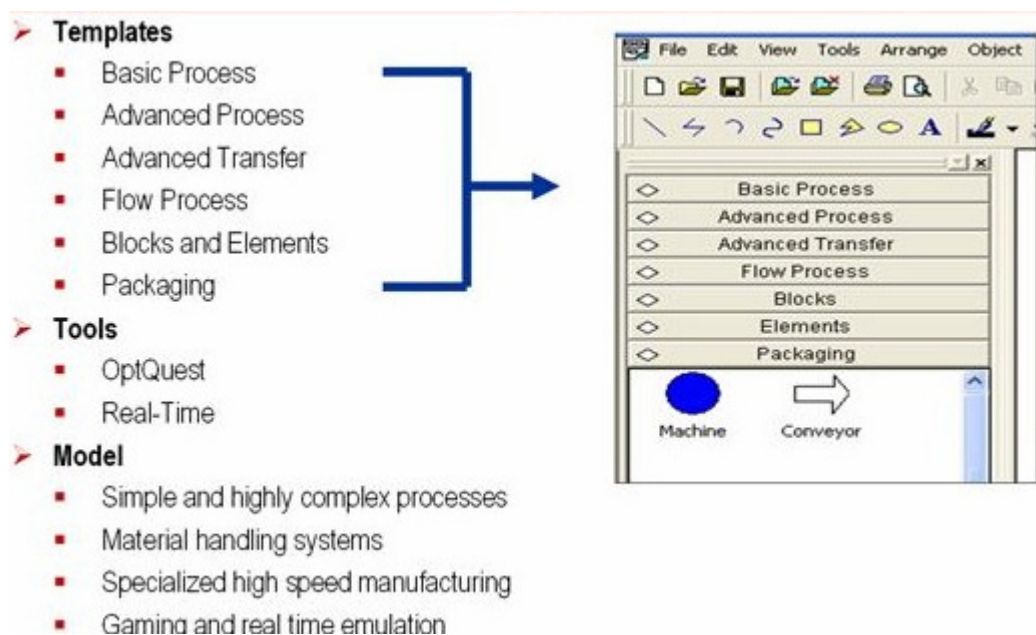


Figura 2-11: Arena Factory Analyser de “ACT Solutions”

- **Arena Contact Center:** biblioteca que permite modelar, especificamente, a dinâmica e a variabilidade inerente aos centros de chamadas.

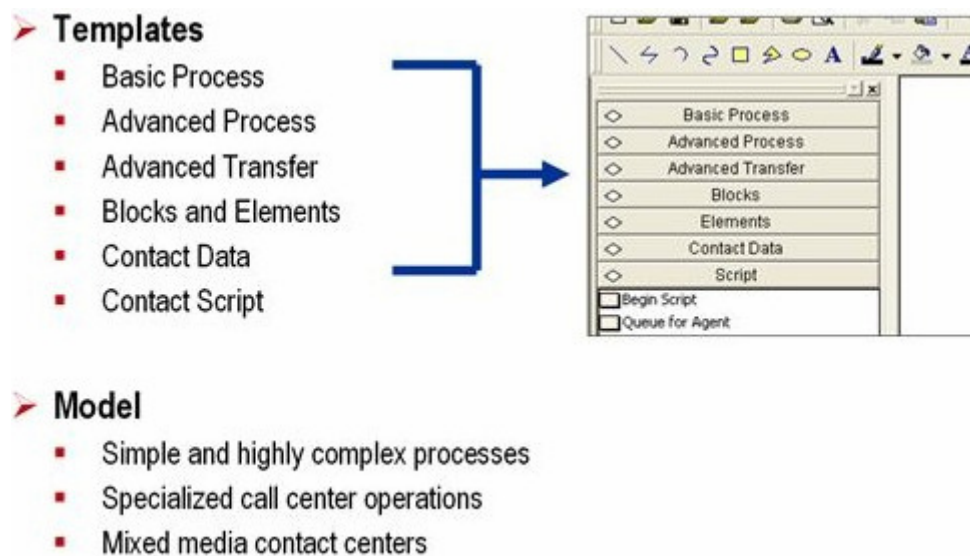


Figura 2-12: Arena Contact Center de “ACT Solutions”

2.10.2.1 O ambiente de simulação Arena

Os dados da entrada de um sistema de simulação correspondem às informações do cenário real a serem submetidas a um modelo. Em muitos casos a obtenção e análise destes dados é uma das partes de um projecto de simulação. Simplificadamente, as etapas de um projecto de simulação são:

- Obtenção e análise de dados de entrada
- Construção do modelo
- Refinamento do modelo
- Análise e apresentação dos resultados

O ambiente de simulação Arena é composto por um conjunto de ferramentas/”software” que possuem funções específicas que se descrevem, sucintamente, de seguida:

1- Simulador: O ‘software’ Arena fornece ao utilizador, o ambiente para o desenvolvimento do modelo de simulação, executa as simulações e gera relatórios apresentando os resultados das experimentações. O ambiente em si, com interface padrão Windows, é destinado à construção, execução e manipulação de modelos. Possui, entre outros, elementos de animação, modelação e interacção com outras aplicações relevantes.

2- “Input Analyzer”: ferramenta utilizada para determinar a qualidade do “casamento” entre as funções de distribuição de probabilidade, com de entrada, recolhidos do sistema real (Araújo et al. (2006)). Esta ferramenta trata os dados de entrada no modelo de simulação e auxilia o analista na determinação da distribuição de probabilidade que melhor representa os dados recolhidos no sistema real.

Uma vantagem que os simuladores possuem, é a possibilidade de traduzir a variabilidade dos eventos na análise dos sistemas. O “Input Analyser” lê os dados recolhidos do sistema real e apresenta histogramas com os dados agrupados por frequências.

Os dados que são processados pelo “Input Analyser” são normalmente os intervalos de tempo entre começo e fim de processamento, ou entre chegadas de entidades aos sistemas e outros sistemas aleatórios (Araújo et al. (2006)). O “Input Analyser” descreve o comportamento do sistema através de equações matemáticas depois de receber um ficheiro do tipo texto, com os intervalos de perturbações e/ou suas durações. Para preparar o ficheiro a ser analisado pelo “Input Analyser”, pode ser utilizado qualquer editor de texto. As distribuições de probabilidade utilizadas pelo “Input Analyser” são: Beta, Contínua, Discreta, Erlang, Exponencial, Gamma, Johnson, Lognormal, Normal, Poisson, Triangular, Uniform e Weibull (Kelton e al. (2002)).

3- “Output Analyzer”: executa a realiza a análise estatística dos dados recolhidos nos ficheiros de saída do ‘software’ Arena.

O analista tem a possibilidade de pedir ao Arena que quando execute o modelo recolha dados para posteriormente apresentar em forma de relatório estatístico quando finaliza a execução do modelo. Se o analista pretender uma análise mais detalhada de determinados parâmetros, é possível gerar um ficheiro adicional, que regista o comportamento desses parâmetros ao longo de todo o tempo de execução da simulação para ser analisado pela ferramenta “Output Analyzer”.

As opções gráficas do programa incluem desenhos, correlogramas, histogramas e outros. Permite a análise de características tais como intervalos de confiança, análise de variância de factor único e comparação de múltiplos sistemas (Takus (1997)). Permite ainda a exportação para ficheiros de texto padrão para serem posteriormente analisados por outras aplicações de estatística.

4- “Process Analyser”: teste e comparação de alternativas, por comparação de vários cenários dos modelos. Uma ferramenta útil para o analista e para quem tem de tomar decisões.

Esta é uma ferramenta que permite ao analista avaliar as alternativas apresentadas para a execução da simulação nos diferentes cenários do modelo do sistema estudado. A ajuda é bastante útil, tanto para quem programa porque permite analisar um conjunto de cenários alternativos, como para quem toma decisões finais que muitas vezes são pessoas que não estão familiarizados com a modelação, sim com os resultados que a simulação oferece. O “Process Analyser” é utilizado na comparação de cenários já depois do sistema conceptual ter sido traduzido para um modelo computadorizado. Nesta fase já o modelo está completo, verificado e validado e

configurado para ser utilizado no “Process Analyzer”. Em conclusão poderemos dizer que é uma ferramenta que permite a comparação entre dois resultados tirados do modelo a partir de diferentes dados na entrada (Araújo et al. (2006)). Foi uma ferramenta muito útil para as conclusões do presente estudo.

5- “Scenario Manager”: com uma configuração fácil num ambiente integrado gere a execução de múltiplas execuções de simulação. O “Scenario Manager” permite que o utilizador especifique múltiplas execuções (cenários) de um modelo para serem executados sem animações. Por exemplo, uma vez que foi desenvolvido um modelo que represente fielmente um sistema do utilizador, é muitas vezes útil saber os efeitos que determinadas entradas podem provocar no modelo. As saídas das execuções dos modelos podem depois ser analisadas através de relatórios ou utilizando o “Output Analyser”, (Takus (1997))

6- Fonte de dados externa: muitos problemas de simulação exigem que se efectuem alterações em dados do modelo para avaliar modificações no cenário. Noutros, necessitamos testar o funcionamento de um sistema para diferentes quantidades de estações de trabalho, ou de servidores, ou de rotas, ou de ‘lay-out’, etc. Para evitar criar diversos modelos de simulação para atender a cada uma das solicitações, é prática corrente criar um único modelo de simulação, suficientemente amplo para comportar o cenário mais ‘pesado’ e utilizá-lo com dados diferentes para analisar variações de funcionamento do cenário em questão (Prado(2004)). O Arena tem componentes “Read” e “Write” localizados no painel “Support” do “template” Arena, que permitem a leitura e escrita em ficheiros de texto, assim como a utilização do VBA para a leitura e escrita de ficheiros em formato de texto. Tem também estruturas de acesso a base de dados do tipo “Data Access Objects” (DAO) que permite aceder e guardar informação em Base de Dados como “SQL Server” ou documentos da “Microsoft Office”.

7- “ActiveX Automation”: esta tecnologia permite que as aplicações se controlem a elas mesmas através de uma interface programada. Através desta tecnologia as acções são automatizadas, permite por exemplo a abertura e fecho de documentos, início e fim de simulação e a integração com outras aplicações projectadas com este fim (Kelton (2002)). É uma tecnologia desenvolvida pela Microsoft para partilha de informação com outras diferentes aplicações.

8- A animação

A animação dos modelos de simulação: num modelo de animação mostra a parte visual do modelo computacional quando este está em funcionamento. A animação mostra ao longo do tempo de execução, o fluxo das entidades no sistema. A animação é cada vez mais uma importante ferramenta na aplicação de modelos de simulação de sistemas. O analista pode visualizar e inspeccionar no monitor de um computador os seus modelos. A representação dinâmica da execução do modelo, permite fornecer ao analista informações importantes sob o desempenho do modelo, que dificilmente seriam obtidas por uma análise estatística dos resultados (Pedgen, et tal. (1990)).

Num processo de simulação, com a utilização da animação o analista poderá realizar um acompanhamento mais pormenorizado do funcionamento do sistema, porque a introdução de determinadas características visuais, permite uma mais fácil

visualização e identificação de tudo o que se passa em qualquer momento da execução da simulação.

No Arena a animação é inserida através de símbolos ou desenhos dos módulos incluídos nos “templates”, podendo ser alterados através de ferramentas internas, muito similares a aplicações de desenho. O Arena permite que o utilizador utilize ícones existentes em bibliotecas ou provenientes de aplicações externas.

Animação na verificação do modelo: a verificação do modelo consiste numa etapa do desenvolvimento do modelo em que o programador verifica se o modelo desenvolvido, corresponde ao idealizado. É verificado se o modelo foi construído correctamente. Não detecta se o modelo conceptual está correcto, ele apenas detecta se o modelo computacional tem erros. A detecção de erros lógicos durante a fase de desenvolvimento do modelo computacional é uma fase por vezes complexa e demorada. Nos casos em que os erros passam despercebidos no modelo, leva o analista a tirar conclusões erradas referentes ao desempenho do sistema. Os resultados da simulação levam-nos à tomada de decisões ineficientes e custosas (Law (1991)). Nestas situações a utilização da animação será concerta uma mais-valia, pois permite visualizar de uma forma rápida esses erros lógicos. Como é uma fase em que o analista faz testes exaustivos no simulador, o encurtamento na detecção dos tempos, agiliza a verificação do modelo encurtando significativamente nos custos do desenvolvimento final do modelo. A animação permite ainda através das observações da animação, compreender melhor todo o processo e diminuir a probabilidade de ocorrência de erros.

Animação na validação do modelo: a validação do modelo é uma etapa em que o analista verifica se o modelo desenvolvido, é uma representação adequada do sistema real. A validação normalmente é conseguida executando o modelo e comparando com sistema real. Se os resultados da simulação se aproximarem dos valores reais, com um nível de confiança desejado, o simulador será validado. A animação evidencia pormenores que o analista e o cliente considerem interessante visualizar. Facilita também a explicação do modelo aos clientes e utilizadores do modelo de simulação, que conhecem o sistema mas podem não conhecer bem a modelação ou simulação. Ambos, analista e cliente poderão através da animação da simulação conjuntamente com alguns gráficos verificar se o sistema foi modelado correctamente.

Animação nas interacções dinâmicas: a animação contribui para o analista melhorar o conhecimento da funcionalidade do sistema e permite verificar de uma maneira mais fácil as interacções entre os componentes desse sistema. Considerando que os métodos estatísticos têm limitações na estimação e comparação do desempenho, a animação pode mostrar ao analista os processos que estão a interagir e qual o seu desempenho. A animação poderá proporcionar melhoramentos, que o analista poderá testar antes da entrega ao cliente final. Os melhoramentos frequentes não são percebidos com uma análise gráfica e estatística das variáveis. Mas com uma visualização no monitor da animação poderão tornar-se óbvias (Paiva (2005)).

Animação na apresentação dos resultados do modelo: cada vez mais nas simulações, a apresentação do desenho gráfico é importante. A animação ilustra o impacto que determinadas decisões têm nos sistemas estudados, diminuindo nos clientes, menos familiarizados com a simulação, alguma resistência que possa haver.

Os gestores, agentes de decisão sobre o sistema real, devem usar os resultados do modelo para que a simulação alcance os efeitos desejados. Uma parte bastante importante de qualquer projecto de simulação é a forma como os resultados dos modelos são apresentados aos agentes tomadores das decisões (Paiva (2005)). Segundo Paiva (2005) "O objectivo básico de qualquer esforço de modelação é prover informação à gestão, que tomará a decisão. Para isto é fundamental que a informação seja confiável (Poorte (1989))".

2.10.2.2 A utilização do VBA no Arena:

‘Visual Basic for Applications’ (VBA) é uma implementação da Microsoft Visual Basic construído para todas as aplicações da Microsoft Office, como outras aplicações como Visio e algumas outras aplicações tais como ‘AUTOCAD’ e ‘MSWord’. O VBA promove uma completa integração com ambiente de desenvolvimento (IDE) familiar aos programadores usando ‘Microsoft Visual Basic’, incluindo um écran de projecto, écran de propriedades e ferramenta de ‘debug’. VBA inclui também suporte para ‘Microsoft forms’, criação de caixas de diálogo, e controlos ActiveX, para que rapidamente seja possível construir interface para o utilizador (Yücel (2005)).

A utilização das duas tecnologias “ActiveX Automation” e VBA juntas, permite ao Arena integrar-se com outros programas que suportam “ActiveX Automation”. O programador pode escrever código “Visual Basic” directamente no Arena, utilizando o editor do “Visual Basic”. O código escrito é guardado no mesmo ficheiro que é guardado o modelo com toda a lógica, dados e animação.

O VBA pode ser usado quando o projecto modelo está a ser lido, executado, terminado ou enquanto as entidades fluem através dos módulos do modelo do Arena.

A utilização do editor de VBA torna o desenvolvimento do modelo mais fácil, porque a verificação da sintaxe de cada instrução de código utilizado, é feita no momento, diminui também consideravelmente o tempo de desenvolvimento e erros indesejáveis. Utilizando o editor é possível correr o programa passo-a-passo e visualizar rapidamente e facilmente as variáveis desejáveis, inclusive as variáveis utilizadas na linguagem SIMAN da simulação.

Pelo facto do código VBA ser independente do Arena, permite ao programador utilizar variáveis que não existem no Arena, facilitando o desenvolvimento do modelo de simulação.

De uma maneira geral, para simular qualquer sistema no Arena é necessário escrever:

- As estações de trabalho;
- O fluxo dentro do sistema;
- As durações, distâncias, velocidades, etc.

A programação visual é a técnica usada pelo Arena para construir qualquer modelo, o fluxo do sistema é criado no écran do computador na forma de um fluxograma, correspondendo à ocorrência de eventos a um cliente genérico, também chamado ‘entidade’ que flui pelo sistema modelado (Prado (2004)). Cada bloco do fluxograma representa um evento no sistema, como por exemplo a chegada de um contentor. A cada evento corresponde um módulo do Arena, os vários módulos ligados entre si, numa determinada sequência formam o fluxograma ou diagrama de blocos.

2.10.2.3 Módulos do Arena:

Um fluxograma é construído a partir dos módulos do Arena. Estes estão disponíveis nas diversas ‘templates’. O Arena Professional versão 10.0 utilizada para o caso de estudo tem disponibilizado as seguintes ‘templates’:

- ‘Advanced Process’
- ‘Advanced Transfer’
- ‘Basic Process’
- ‘Blocks’
- ‘CSUtil’
- ‘Elements’
- ‘FlowProcess’
- ‘FlowProcessUtil’

Para que o sistema continue a ter compatibilidade com as versões anteriores, estão disponibilizadas também as ‘templates’ das versões anteriores à versão 4.

- Common;
- Support;
- Transfer

Os módulos de uma ‘template’ dividem-se em duas categorias, ver Figura 2-13 referente à ‘template Basic Process’:

- Módulos do fluxograma: Utilizados para construir o fluxograma ou diagrama de blocos dentro da área de trabalho (área utilizada para colocar o fluxograma com a lógica do modelo).
- Módulo de dados: Recebem dados provenientes do modelo, mas não são colocados dentro da janela de trabalho.

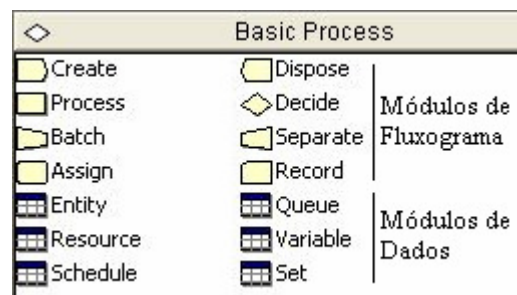


Figura 2-13: Módulos da 'template' Basic Process

3 Caso de Estudo

3.1 Introdução

No capítulo anterior foi realizada uma introdução à simulação e explicou-se a razão da utilização da aplicação de simulação Arena no desenvolvimento do modelo pretendido. Este capítulo apresenta o sistema de Logística Interna da empresa Codizo que serviu de caso de estudo para este trabalho.

Um dos principais problemas que a indústria de calçado tem vindo a sentir é que os mercados cada vez mais procuram lotes de menores dimensões e com prazos de entrega mais curtos. As indústrias anteriormente estavam mais ajustadas a lotes de grandes dimensões e por isso, as metodologias que funcionavam eficientemente deixaram de o fazer. Há necessidade de preparar as indústrias para responder melhor a estes caprichos do mercado. Essa preparação passa entre várias coisas por uma melhor organização interna, como por exemplo, tornar as implantações fabris ‘layouts’ mais racionais. A distribuição física dos equipamentos pode não estar adequada às quantidades e à diversidade de produtos produzidos. Poderá haver custos constantes de movimentação de materiais e pessoas que poderiam ser minimizados. Orientar a produção ao produto, quer com linhas quer com células é a solução frequente. Implantações mais racionais trazem normalmente ganhos significativos de produtividade. Se observarmos o dia-a-dia da planta fabril de uma empresa, verificamos a existência de uma permanente movimentação “desorientada” e desorganizada de pessoas, materiais e produtos. Se analisarmos as causas dessas movimentações, constata-se que tal se deve sobretudo à cada vez maior dificuldade de implementar processos organizacionais e procedimentos mais consistentes, que permitam aos operadores organizar criteriosamente os planos de produção, dado o crescente volume de encomendas muito pequenas de modelos distintos que actualmente caracterizam o sector do calçado.

A empresa de calçado Codizo, resolveu minimizar os custos da constante movimentação de materiais e pessoas implantando um sistema de movimentação completamente automático que permitisse então o transporte do produto interno entre as suas secções de produção. O sistema de logística interno na produção começou então a ser implementado no ano 2005 e ficou em pleno funcionamento em meados de 2006.

O sistema desenvolvido e instalado na empresa é composto por um sistema automático de armazenamento intermédio do produto em curso de fabrico e componentes, de uma forma agregada para cada uma das secções da empresa. Este sistema de armazenamento está dividido em quatro armazéns com uma capacidade total de 8880 pares, considerando que cada caixa poderá levar 20 pares. É um sistema

automático de transporte composto por uma vasta rede de tapetes e torres elevatórias que efectua a interface entre os armazéns e as seis secções de fabrico.

Todo o sistema de logística interna é gerido por um sistema de gestão operacional e por um sistema de automação desenvolvido pelo INESC Porto e pela empresa Lirel. O INESC Porto desenvolveu o sistema de gestão operacional e toda a automação do sistema de logística interna (tapetes mais torres), a Lirel desenvolveu a estrutura do sistema e a automação dos armazéns de produto interno.

A gestão do atendimento das secções de montagem e a elaboração das rotas de produção está a cargo do módulo de gestão operacional. Este módulo permite aos operadores, de uma forma intuitiva, otimizar a produção de cada uma das secções de fabrico e determinar automaticamente a melhor rota para cada uma das caixas das ordens de produção. Reflectir na planta fabril as decisões do planeamento é imediato, permitindo uma melhor utilização de todos os recursos disponíveis, reduzindo o tempo em produção das encomendas e um melhor cumprimento dos prazos acordados com os clientes.

O modelo em estudo pretende ser o modelo de simulação, que representará o sistema de logística interna implementado na empresa Codizo, ou seja, o sistema de movimentação de contentores (caixas) com produto interno, entre várias secções de produção da empresa.

3.2 A empresa

Na empresa Codizo, o layout está dividido em várias secções e o produto interno sofre alterações nas diversas secções conforme a gama operatória previamente definida. Para o sistema de movimentação de caixas em estudo, só se considerou a existência na empresa, de duas grandes secções: costura e montagem, sendo esta última a mais focada.

3.2.1 Caixas

O tipo de caixas utilizadas para o transporte do produto interno em curso é mostrado na Figura 3-1. Estas foram escolhidas tendo em conta: número de pares capazes de transportar, a acomodação dos materiais, optimização da ocupação do espaço interior do contentor, a sobreposição de caixas sem danificar o seu conteúdo, a fácil manipulação por sistemas mecânicos e a sua resistência.



Figura 3-1: Contentor utilizado para o transporte de produto interno

O número de pares depende dos modelos. No transporte utilizam-se caixas com 2 pares até 20 pares

3.2.2 Secções

Na parte produtiva da empresa existem 5 secções principais: Corte, Pré-costura, Costura, Pré-montagem e Montagem. Para além da produção existem outras secções: armazenamento de matéria-prima e materiais e armazenamento de produto acabado.

Secção de armazenagem de matéria-prima e materiais: nesta secção estão armazenadas as matérias-primas a utilizar na empresa. O trabalho aqui realizado é normalmente manual e conforme as encomendas, os materiais são separados nas quantidades adequadas, e são transportados manualmente até às secções.

Secção de Corte: o corte das peles ou sintéticos a utilizar no calçado, pode ser realizado automaticamente utilizando por exemplo máquinas de corte por jacto de água, ou manualmente, com ajuda de prensas e cortantes. As peles ou sintéticos denominados por gáspeas são cortados, separados e agrupados por tamanhos. As gáspeas são reagrupadas por lotes e colocadas em caixas. Cada contentor poderá ter um ou mais lotes.

Secção de Pré-costura: as gáspeas chegam dentro das caixas a cada operador na secção, o operador separa novamente, realiza a operação da gama operatória e volta a colocar no contentor, passando para o operador que realiza a operação seguinte. As gáspeas voltam a sair da secção dentro das caixas para seguirem para a secção de costura.

Secção de Costura: o percurso das gáspeas é similar ao da secção de pré-costura. As gáspeas voltam a sair da secção dentro das caixas para a pré-montagem.

Secção de Pré-montagem: as gáspeas chegam dentro das caixas e aqui sofrem um conjunto de alterações em vários postos de trabalho, preparando-as para seguirem para a secção de montagem. As gáspeas voltam a sair da secção dentro das caixas.

Secção de Montagem: as gáspeas chegam dentro das caixas, são separadas e colocadas num transportador que se movimenta em círculo, a uma cadência predefinida (imposta pelo número de pares a produzir por dia) passando por vários postos sofrendo as várias operações da gama, ou seja, aqui as gáspeas sofrem um conjunto de operações, uma delas por exemplo é a união com as solas tomando a forma definitiva. As caixas ficam vazias e prontas a serem reutilizadas com outras gamas operatórias.

Secção de armazenamento de produto acabado: nesta secção o produto em curso aguarda até ser enviado para os clientes. Os pares são armazenados aos pares em cada caixa, e estas são colocadas em tarifas de 25 pares e agrupadas em paletes de 25 tarifas.

SubContratacção: devido à capacidade de produção das três linhas de montagem ser superior à da linha de costura, algumas das gáspeas sofrem alterações fora da empresa, nos subcontratados. Existem casos em que o corte e costura são realizados fora da empresa, e posteriormente a montagem na empresa. Noutros casos, o corte é realizado na empresa, e a costura fora da empresa e posteriormente voltam a entrar para realizarem as restantes operações. Em ambas as situações as gáspeas são montadas na Codizo, ou seja, têm de entrar no sistema, ou voltar a entrar no sistema para sofrerem as operações de montagem.

O sistema em estudo só realiza o movimento entre as secções de costura, pré-montagem e montagem. A secção de costura é composta por uma linha de costura com cerca de oito operadores. A secção de pré-montagem é composta por uma linha de montagem com cerca de cinco operadores. A secção de montagem está dividida em três linhas independentes chamadas Montagem I com cerca de vinte e seis operadores, Montagem II com cerca de vinte e seis operadores e Montagem III com cerca de quinze operadores. A capacidade de produção das três montagens é superior à capacidade de produção das secções de costura. Por esta razão a empresa recorre à subcontratação da produção de costura.

Devido à particularidade de alguns postos de trabalho como o posto de moldar na secção de pré-montagem, o posto de enformar, na linha de montagem III da secção de montagem e a subcontratação, foi necessário considerar no sistema de movimentação, também entrada e saída de caixas nesses locais, tal como nas restantes secções.

Moldar: as gáspeas chegam dentro das caixas, sofrem as operações de moldar (ganham forma) numa máquina, um operador, e voltam a sair da secção dentro das caixas para realizarem as operações seguintes.

Enformar: as gáspeas chegam dentro das caixas, sofrem as operações de enformar (ganham forma numa máquina, um operador, e voltam a sair da secção dentro das caixas.

Na Figura 3-2 estão representados os locais da Entrada/Saída das caixas no sistema produtivo da empresa

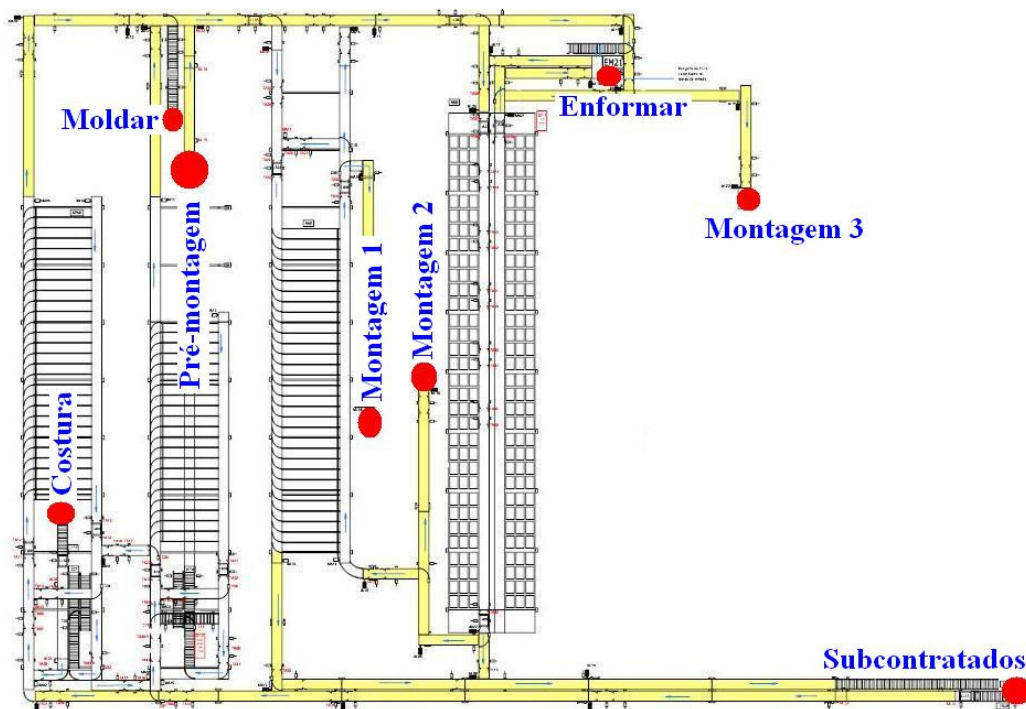


Figura 3-2: Secções no Sistema de armazenamento e movimentação instalado

3.2.3 Gamas operatórias

As gamas operatórias na empresa podem ser vistas sob ponto de vista macro ou micro. Sob ponto de vista macro, consiste na sequência de secções a percorrer ao longo do sistema produtivo. Sob ponto de vista micro, consiste na sequência de operações a realizar dentro de uma secção, por exemplo o caso mais evidente é a secção de costura. Aqui os pares podem sofrer dezenas de operações, e neste caso, também é indicado, o posto e máquina onde essa operação é realizada, além do tempo previsto para a sua realização. No sistema de movimentação em estudo, só foi considerada a gama operatória macro, porque só essa é necessária.

No sistema de movimentação considerado, existe uma gama operatória para cada contentor, indicando só as secções do sistema produtivo por onde o contentor terá que passar. O sistema informático responsável pela gestão operacional importa dum ERP, informação sobre os Modelos, as Ordens de Fabrico e os Lotes e envia a gama operatória e as respectivas configurações para o PLC responsável pela automação da logística interna (movimentação). As gamas operatórias dos modelos são criadas pelo operador antes das caixas entrarem no sistema de movimentação ou, no pior dos casos, no instante de associação do primeiro contentor do modelo que vai entrar no sistema, ver Figura 3-3. O operador responsável pela tarefa de associação de caixas ao sistema, graficamente cria a gama do modelo, indicando os locais por onde todas as caixas com esse modelo, terão que passar, conforme se pode verificar na Figura 3-3, o operador tem do lado esquerdo em baixo, todas as secções da empresa premindo duas vezes seguidas na secção pretendida, ou seleccionando e premindo no botão inferior (→) ela passa para o lado direito. No lado direito está representada a rota por onde passarão todas as caixas do modelo seleccionado.

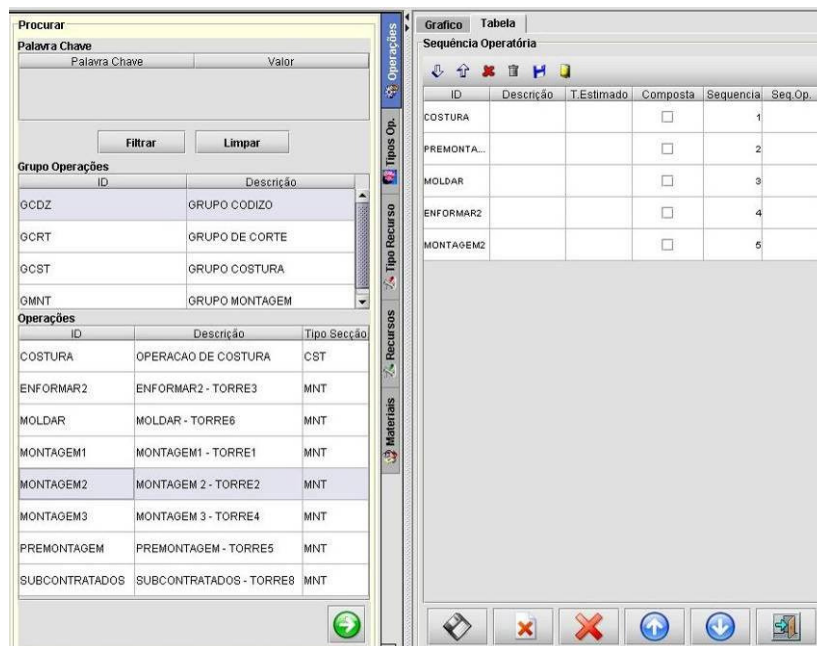


Figura 3-3: Construção da gama operatória através de tabela

O operador tem também a possibilidade de construir a gama operatória de um modo gráfico conforme mostra a Figura 3-4, escolhendo se quer visualizar a gama em modo gráfico ou em modo de tabela. No modo gráfico poderá arrastar as operações (objectos) para sequência desejada.

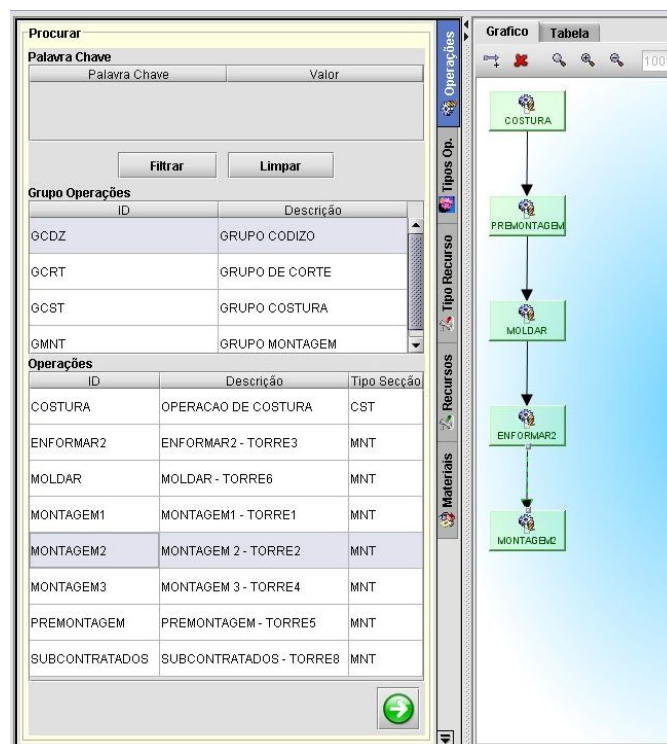


Figura 3-4: Construção da gama operatória graficamente

A cada secção da gama operatória é atribuída um código de 0 a 99 que é enviado para o PLC responsável pela gestão da movimentação dos tapetes. O PLC recebe então por cada contentor, um conjunto de códigos na sequência predefinida pelo operador que criou a gama operatória. O PLC só deixa entrar caixas no sistema de movimentação se essas caixas tiverem informação, ou seja se tiverem uma gama operatória. Sempre que um contentor entre numa secção o PLC “abate” a operação “seguinte” ainda por realizar, e quando é realizada a última operação do contentor, este é abatido, e ficará pronto para receber novamente informação.

3.2.4 Sistema de gestão operacional

O sistema de gestão responsável pela operação é composto por uma rede de PC's, utilizando uma arquitectura distribuída *cliente servidor*, ver Figura 3-5. Os clientes estão localizados junto aos locais onde há necessidade de entrada e saída de caixas, conforme Figura 3-5.

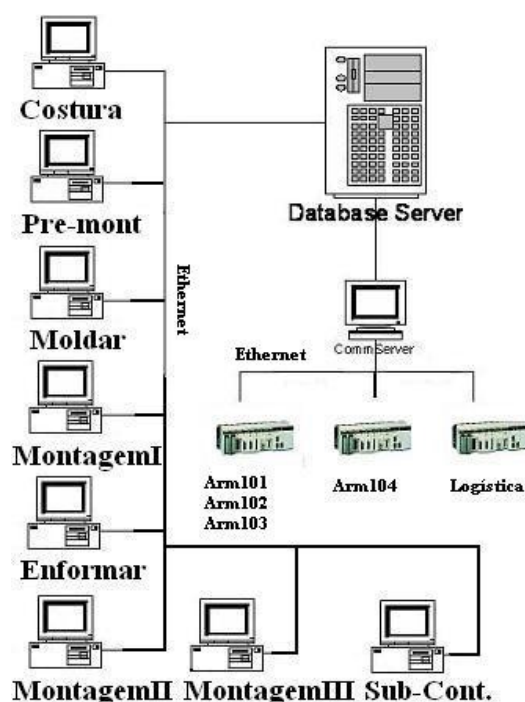


Figura 3-5: Arquitectura cliente-servidor

Na arquitectura *cliente servidor*, os clientes estão localizados em quiosques, em locais estratégicos das secções, ou seja, em todos os locais onde é necessário entrar ou sair caixas do sistema de movimentação conforme se pode ver na Figura 3-6, permitindo que os operadores das secções, interagem com todo o sistema, de uma maneira simples, através de écrans tácteis.

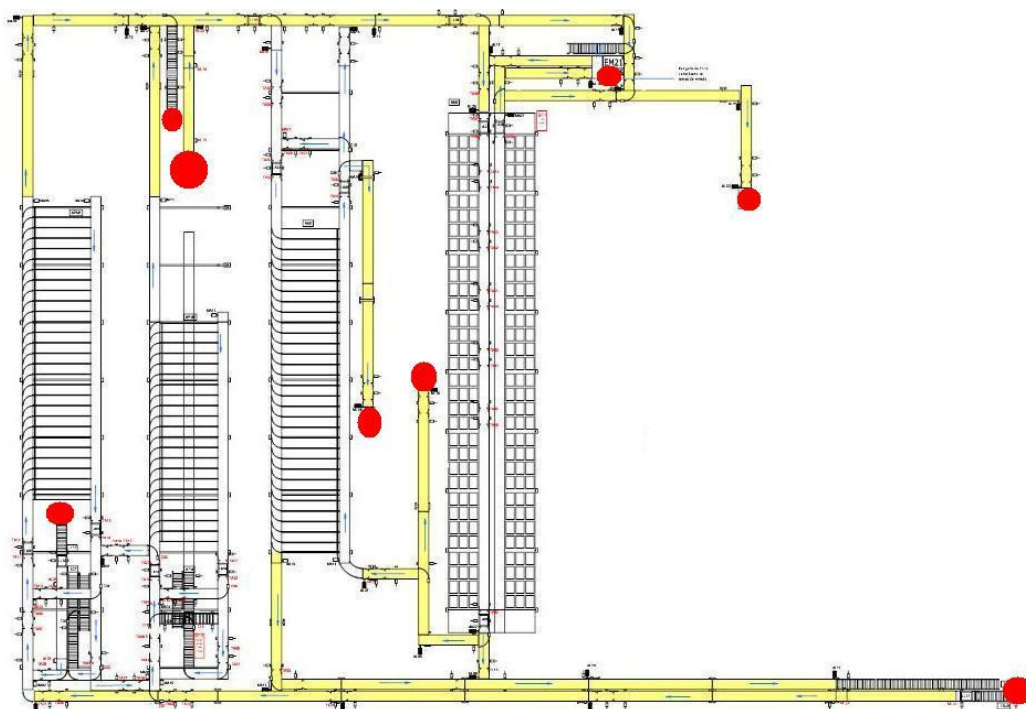


Figura 3-6: Locais de entrada/saída de caixas

3.2.5 Sistema de automação

Todo o sistema de automação é realizado através de três PLC's, ver Figura 3-5. Dois deles realizam a automação referente aos quatro armazéns, e o terceiro realiza toda a automação referente aos tapetes e toda a gestão da movimentação entre secções e armazéns e vice-versa. A interface entre o sistema de gestão operacional (PC's) e o sistema de automação (PLC's) é realizada por uma aplicação chamada "Commserver" residente no mesmo computador onde está o servidor. A aplicação Commserver desenvolvida em Visual Basic que utiliza um Componente com o protocolo de comunicação utilizado para comunicar com os PLC's e utiliza "Sockets" para comunicar com os clientes PC's, conforme Figura 3-7.

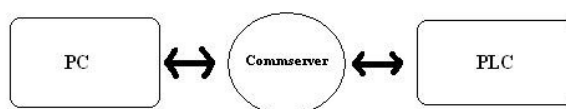


Figura 3-7: Interface PC - PLC

O sistema de gestão operacional comunica com todos os PLC's sempre que existe necessidade de monitorizar partes do sistema, mas no que se refere à gestão da movimentação comunica só com o PLC responsável pela movimentação dos tapetes, enviando para ele, as gamas operatórias, configurações, e recebe deste, relatórios, tais como por exemplo, o registo do nº do contentor, data e hora de chegada e saída do

mesmo, numa secção. Mais à frente serão descritos mais em pormenor os envios e recepções de dados dos PLC's. Toda a informação das caixas necessária à movimentação está residente no PLC que gere a movimentação dos tapetes, chamado PLC da Logística. Este tem uma grande capacidade de memória para armazenar uma elevada quantidade de dados, sendo possível ter alguma autonomia de tempo em caso de quebra de comunicação entre os PLC – Commserver – PC Cliente.

Os dois PLC's responsáveis pela gestão dos quatro armazéns gerem toda a movimentação nos armazéns de uma maneira completamente autónoma do resto do sistema. As comunicações entre os PLC's responsáveis pelos armazéns e o PLC responsável pela logística interna são realizados utilizando o meio físico Ethernet e um protocolo definido entre ambas as partes (armazéns e logística) que é utilizado sempre que um contentor chega ou sai dos armazéns e sempre que é necessário monitorizar alguma variável.

3.2.6 Constituição do sistema de movimentação e armazenamento

Os sistemas automáticos de armazenagem existentes no mercado não se enquadravam com a diversidade e desempenho requeridas por este sistema automático de armazenamento. O sistema desenvolvido para a empresa tem exigências ao nível da velocidade e especificidades de configuração de postos de associação de materiais, não encontradas noutros sistemas de armazenagem.

Antes da instalação do sistema automático de transporte, a movimentação das caixas entre secções era efectuada manualmente com a dificuldade, erros e perda de produtividade que estavam inerentemente associadas a este tipo de movimentação.

A diversidade dos modos de funcionamento e dos sistemas de gestão e logística das secções produtivas das fábricas de calçado fazia com que não existisse no mercado nenhum sistema integrado de transporte e distribuição entre secções. Existiam no entanto casos isolados de projectos de integração entre duas secções.

Algumas funcionalidades do sistema de movimentação:

- Efectuar a integração com sistemas de armazenagem automáticos.
- Transportar as caixas com os lotes das ordens de fabrico entre secções através de um sistema de tapetes com configuração modular e escalável de modo a poder adaptar-se a qualquer 'layout' fabril.
- Sistema elevatório para a colocação e retirada das caixas das secções para o sistema transportados e em sentido contrário e deverá possuir pequenos "buffers" de entrada e saída de caixas. A sua concepção é independente dos sistemas de tapetes para que possam ser ligadas a qualquer ponto deste, permitindo assim as mudanças de layout com o mínimo de esforço.

- Este sistema foi construído de modo a não interferir com a normal circulação de máquinas, materiais e pessoas dentro da fábrica.
- Gestão de percursos com optimização do fluxo de caixas.
- Possibilidade de ter modo automático (seguindo as gamas operatórias fornecidas pelo ERP)
- Integração com sistema de gestão operacional das secções.
- Sistema de movimentação com configuração modular e escalável de modo a adaptar-se a qualquer layout fabril.
- Definição de prioridade de ordens de produção, de forma a tornar mais eficiente o funcionamento global do sistema.
- Transporta caixas com os lotes das ordens de fabrico directamente entre as secções produtivas da planta fabril
- Efectua a integração das secções produtivas com sistemas de armazenagem automáticos movimentando caixas
- Este sistema integra-se com um grupo de secções com propriedades bem distintas, secção de costura e montagem. Cada uma destas secções requer um conjunto de componentes diferentes quanto à ocupação de espaço, peso, etc.
- A integração com as secções produtivas é feita directamente com um operador, o operador retira as caixas do sistema de movimentação
- A integração também pode ser feita a um equipamento da secção, equipamento automático que executa uma operação e devolve as caixas ao sistema de transporte e distribuição
- Integração com sistema de gestão operacional das secções.

Apesar dos elevados requisitos do sistema, pretendeu-se que o custo da solução final fosse baixo e competitivo, impondo uma boa relação custo funcionalidades. O sistema de movimentação é essencialmente composto por tapetes unidireccionais com cruzamentos, armazéns e torres.

Os **tapetes** transportam caixas com produto interno das secções para os armazéns, dos armazéns para as secções e de armazéns para armazéns. Nos locais em que existe necessidade de cruzamentos, a mudança de direcção é realizada, utilizando cancelas pneumáticas, ver Figura 3-8. A velocidade constante utilizada nos tapetes é de 1m/s.



Figura 3-8: Tapetes utilizados

As **torres** elevatórias (Figura 3-9), têm por missão realizarem a interface entre os tapetes que estão a uma altitude de cerca de 3 metros, com os operadores, nos locais em que não foi possível por falta de espaço, colocar rampas com tapetes. Nos locais onde é necessário as caixas entrarem e voltarem a sair da secção as torres têm 2 ‘buffers’ de caixas, um de entrada de caixas na secção e outro de saída de caixas da secção. Em ambos os casos é necessário um operador, para retirar ou colocar as caixas dos ‘buffers’. O ‘buffer’ superior representa a entrada de caixas na secção e o ‘buffer’ inferior, com rolos motorizados, representa o ‘buffer’ de saída de caixas da secção. Nos locais onde só há recepção de caixas ambos representam buffers de entrada na secção. A capacidade de cada ‘buffer’ nas secções é de 4 caixas.



Figura 3-9: Torre elevatória

Os quatro **armazéns** têm como finalidade, a armazenagem de produto interno intermédio, constituindo parte do ‘stock’ intermédio. Toda a movimentação e armazenamento são realizados no plano horizontal a cerca de três metros do chão-de-fábrica. Desta forma não há interferência entre o sistema de movimentação e armazenamento e o normal funcionamento das máquinas e operadores da empresa. A Figura 3-10 representa o ‘layout’ da empresa em estudo, onde estão representados os 4 armazéns instalados na empresa.

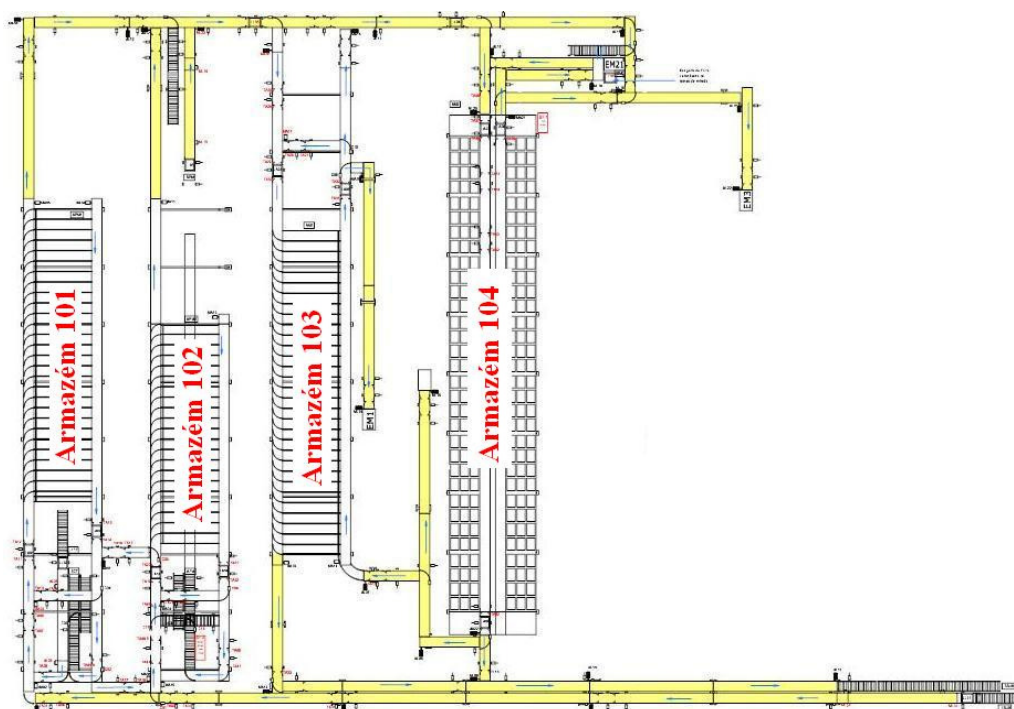


Figura 3-10: Localização dos armazéns da empresa

Existem dois tipos de armazéns: por gravidade, representado na Figura 3-11 e num plano horizontal, representado na Figura 3-14. Os armazéns por gravidade são compostos por módulos de quatro caixas inclinadas e encostadas, conforme se pode verificar na Figura 3-11. Este tipo de armazém, tem a particularidade de sempre que é necessário retirar uma caixa, e no caso em que esta não esteja na fila da frente, e esteja na fila de trás, na quarta fila, que é o pior caso, seja necessário retirar as três caixas da frente, voltar a colocá-las novamente no armazém, antes de retirar a caixa pretendida. Por outras palavras, se a caixa pretendida por uma torre, não estiver na fila da frente, está sempre dependente da saída de outras caixas, que no pior caso, poderá ser necessário a saída de três caixas.



Figura 3-11: Exemplo dos armazéns 101,102 e 103

Contudo o sistema quando retira uma caixa do armazém, não fica à espera que esta chegue ao fim do tapete do armazém, para dar ordem para sair a caixa seguinte, senão tornaria a movimentação de caixas muito lenta. O sistema, por exemplo, caso necessite de retirar uma caixa que está na terceira fila, dá ordem para saírem as três caixas conjuntamente, as duas primeiras quando chegam ao início do armazém, ver

Figura 3-12, retornam ao armazém, e a terceira segue para a torre que a pediu. Na figura as setas representam o sentido de movimentação das caixas. Sempre que é necessário colocar uma caixa no armazém, ela terá que seguir a seta representada no lado esquerdo da figura. Sempre que é necessário retirar uma caixa do armazém, ela terá que deslocar-se no sentido representado na mesma figura do lado direito.

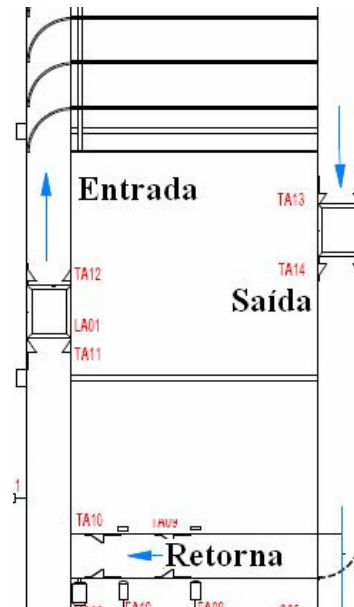


Figura 3-12: Sentido de deslocamento dos Caixas nos armazéns

No armazém num plano horizontal (armazém 104) existe um manipulador de 3 eixos que coloca as caixas provenientes dos tapetes nos alvéolos e vice-versa, conforme a (Figura 3-13).



Figura 3-13: Armazém num plano horizontal (armazém 104)

O armazém 104 contém no seu interior dois tapetes que circulam em sentidos contrários. Ao centro do armazém as caixas têm obrigatoriamente de pararem, porque existe uma barra transversal. É ao centro, que normalmente o braço robotizado pega

nas caixas e as coloca nos respectivos alvéolos, sempre que o objectivo é armazenar o contentor. Se o objectivo é retirar um contentor do armazém, o movimento é realizado no sentido inverso. O sentido de movimentação está representado na (Figura 3-14).

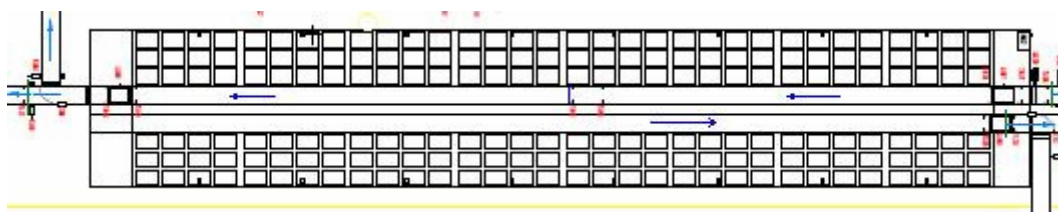


Figura 3-14: Sentido de movimentação das caixas no armazém 104.

3.3 Modo de funcionamento

Apesar de só haver 4 torres elevatórias, (torre 1, torre 2, torre 3 e torre 4) consideram-se os restantes pontos de entrada e saída como torres, porque o modo de funcionamento das 4 torres é igual aos restantes pontos de entrada e saída. Sendo assim, são considerados então 8 pontos de entrada e saída que serão tratadas por torre 1 (T1), torre 2 (T2) ... torre 8 (T8). Na Figura 3-15 está representada a disposição física das 8 torres, sendo bem visível, que a saída da secção da pré-montagem (T5), representada com um círculo, está bem distante da entrada da mesma secção. Nas torres 6, 7 e 8 apesar dos locais de entrada e saída serem em locais diferentes, estes encontram-se muito perto um do outro.

Funcionalidades das torres:

Torre 1 - Montagem I - Esta torre só recebe caixas.

Torre 2 - Montagem II - Esta torre só recebe caixas.

Torre 3 - Montagem 2 Enformar – Esta torre encontra-se no espaço de Montagem 2. Recebe e envia caixas (normalmente para torre2).

Torre 4 - Montagem III – Esta torre só recebe caixas.

Torre 5 - Pré-Montagem – Esta torre recebe e envia caixas. Os locais de recepção e envio estão localizados em espaços diferentes.

Torre 6 - Moldar – Esta torre recebe e envia caixas. Os locais de recepção e envio estão localizados em espaços diferentes. O local de envio é o mesmo da Torre 5.

Torre 7 - Costura - Esta torre recebe e envia caixas. Os locais de recepção e envio estão localizados em espaços diferentes.

Torre 8 - Subcontratados – Esta torre recebe e envia caixas. Os locais de recepção e envio estão localizados em espaços diferentes.

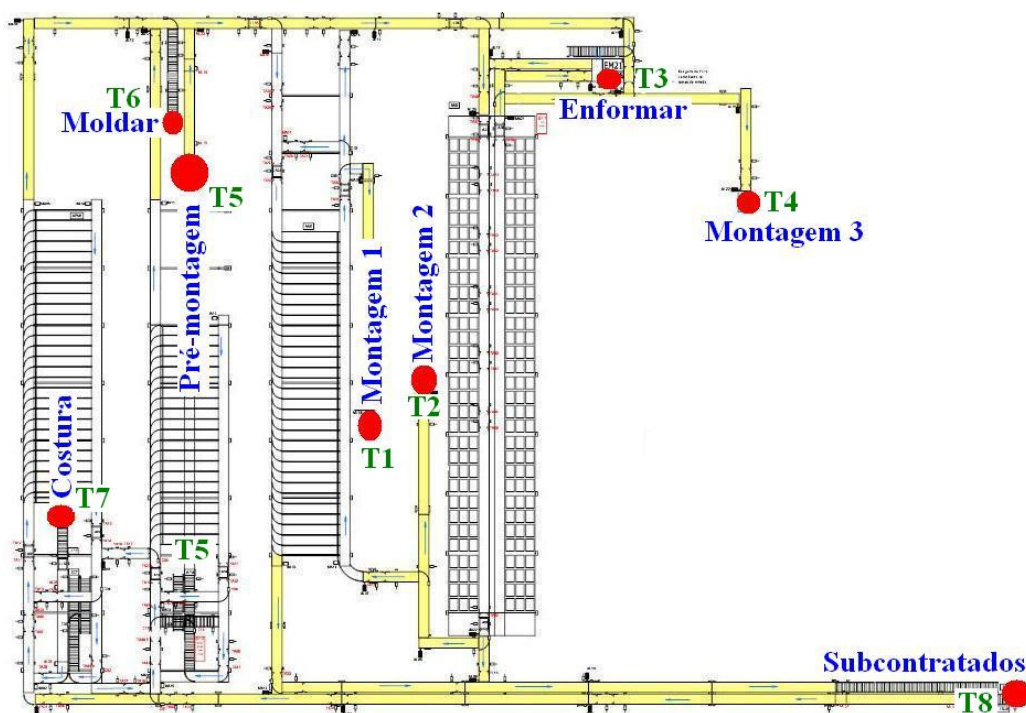


Figura 3-15: Local das torres

O princípio básico do funcionamento de todo o sistema, baseia-se em torres a pedirem caixas aos armazéns, com base num conjunto de critérios explicados mais à frente.

O sistema de gestão operacional considera como secção todos os locais de entrada e/ou saída de caixas, assim deste modo por exemplo, na torre 3, o sistema de gestão operacional considera a secção de “Montagem 2 Enformar” conforme Figura 3-16. O sistema de gestão operacional permite várias configurações para o sistema de movimentação. Através de menus é possível configurar o PLC responsável pela logística, indicando quais os códigos a realizar numa determinada secção. A Figura 3-16 mostra que a secção “Montagem 2 Enformar”, onde está a Torre 3, realiza a operação com código 18.

Figura 3-16: Montagem 2 Enformar (Torre 3)

Cada secção pode ser configurada para pedir caixas aos armazéns de 4 maneiras diferentes: Lista fixa, lista flexível, lista flexível e códigos e por fim só códigos, conforme Figura 3-17.

Figura 3-17: Configuração das listas numa secção

Se o operador escolher **lista fixa**, significa que aquela torre irá pedir caixas aos armazéns pela ordem da lista de caixas definida pelo operador e enviada previamente para o PLC da logística. Neste tipo de configuração as caixas chegam à montagem pela ordem desejada na lista. Se existe algum problema mecânico que impeça a saída da caixa do armazém, o sistema não envia mais nenhuma caixa para a torre até que o problema seja resolvido. Se a próxima caixa da lista não estiver em nenhum armazém

significa que as caixas seguintes, mesmo estando no armazém, nunca serão pedidas. Este modo garante a sequência de caixas programadas pelo operador.

Se o operador escolher **lista flexível**, significa que a torre irá pedir caixas aos armazéns pela ordem da lista de caixas formada pelo operador e previamente enviada para o PLC da logística, mas na altura do pedido se a próxima caixa da lista não estiver em nenhum dos armazéns, mas se estiver a outra a seguir da lista, então será essa a ser pedida, caso a seguinte também não esteja em nenhum armazém, mas esteja a outra a seguir, será então essa outra a ser pedida, ou seja, será sempre a próxima caixa da lista que esteja num dos armazéns. Quando a primeira caixa da lista estiver no armazém será essa a primeira a ser pedida.

Se o operador escolher a **lista flexível e códigos**, significa que a torre tentará pedir caixas aos armazéns conforme descrito anteriormente para a lista flexível, mas caso não encontre nenhuma caixa da lista nos armazéns, ou não exista a dita lista no PLC, o pedido será baseado num código, isto é, em todos os armazéns é verificado qual a caixa com mais prioridade cuja próxima operação seja realizada na torre que está a pedir. Mais à frente será descrito mais em detalhe.

Se o operador escolher **só códigos**, significa que não existe lista e os pedidos são só baseados em códigos conforme descrito anteriormente. Por ser este o caso em que as caixas fluem mais rapidamente no sistema por não estar dependente dos operadores realizarem listas, **foi esta configuração considerada no estudo da simulação.**

As listas são enviadas para a área de memória do PLC de logística, ver Figura 3-18, sendo a mesma área utilizada para todas as listas, ou seja, só é possível ter uma configuração para cada secção num determinado instante.

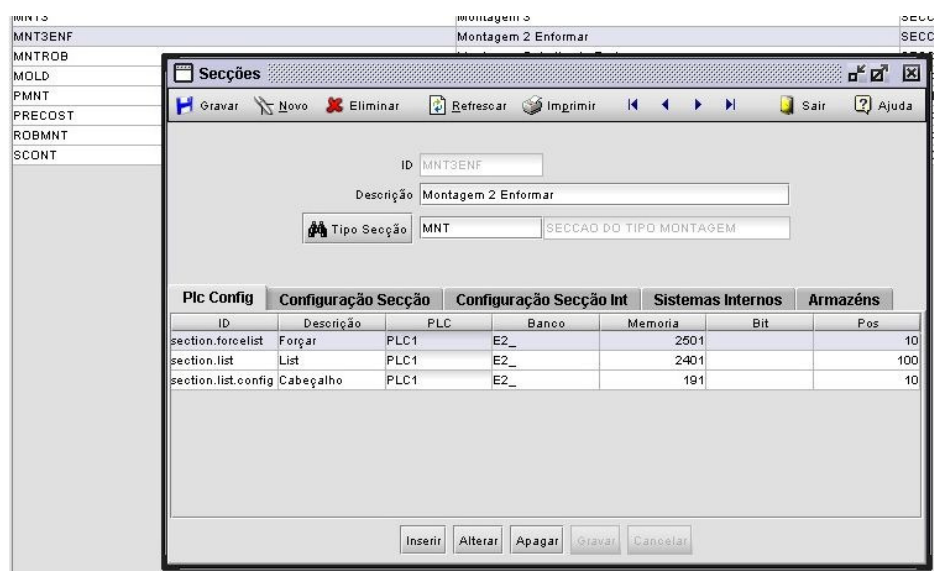


Figura 3-18: Área de memória no PLC onde são escritas as listas de caixas

3.4 Pedido de caixas

Como referido anteriormente a configuração onde poderá haver uma maior quantidade de caixas a deslocarem-se nos tapetes será o caso em que as torres estão configuradas para pedir caixas aos armazéns através de códigos.

As gamas operatórias estão na memória do PLC da logística. A gama operatória de uma caixa é composta por códigos. No exemplo da Figura 3-19 com 6 caixas temos 3 gamas operatórias diferentes. A gama operatória composta pela sequência de códigos A, B e C caixas 1, 4 e 5, a gama composta por E, F e G, caixas 2 e 6 e a gama composta por B, C e D caixa 3. Na figura poderemos ver a '**Bold**' ainda que, a próxima operação a realizar na caixa 1 será a operação com o código B, na caixa 2 será a operação com o código G etc.

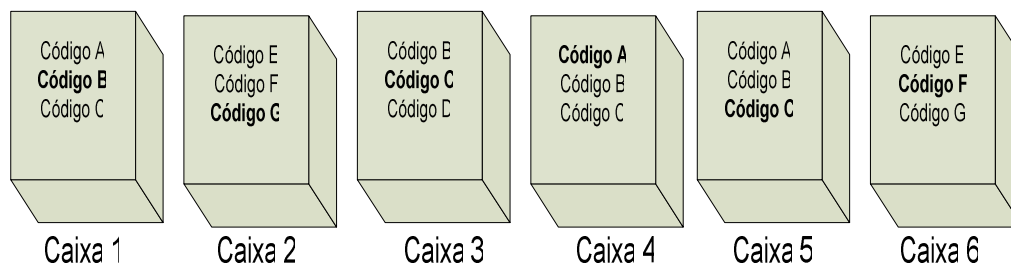


Figura 3-19: Exemplo de gamas operatórias

As torres só podem pedir caixas aos armazéns. Supondo então que as torres realizam os códigos exemplificados na Figura 3-20, e se considerarmos que as caixas referidas anteriormente estão nos armazéns, então para a torre 1 irá a caixa 4; para a torre 2 irá a caixa 6; para a torre 3 irá primeiro a caixa 3, depois a caixa 5 e por fim a caixa 2; para a torre 4 não irá nenhuma caixa, porque a torre faz o código D e não há nenhuma caixa cuja próxima operação será o código D, neste caso a próxima caixa a vir para a torre 4 será a caixa 3 depois de ter realizado a operação com código G numa determinada secção e de ter voltado para os armazéns.

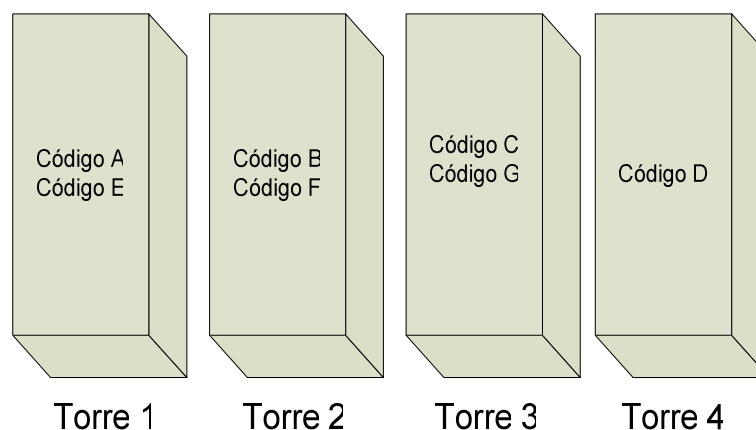


Figura 3-20: Exemplo de gamas operatória

O modo como o PLC responsável pela gestão da logística interna procura as próximas caixas a irem para as torres está ilustrado na Figura 3-21. Exemplificando, a torre 1 primeiro verifica para o código 1 no armazém 101 depois no armazém 102, até ao armazém 104. Depois de procurar nos quatro armazéns encontrou uma caixa, deixa de procurar para essa torre e guarda o local (armazém) e o nº da caixa. Se não encontrou nenhuma caixa em todos os armazéns, verifica o mesmo para o código 2, etc. Depois de a torre 1 ver para todos os códigos será a vez da torre 2 até chegar a vez da última torre.

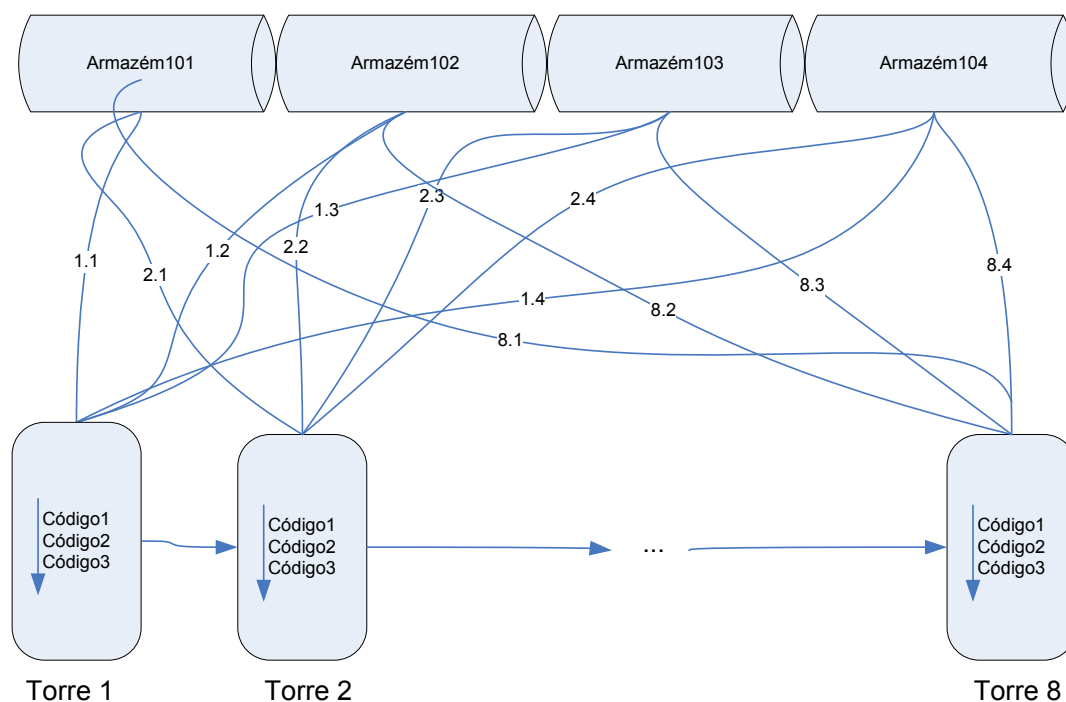


Figura 3-21: Sequência de pedidos de caixas das torres aos armazéns

Na Figura 3-22 está representado o algoritmo que o PLC responsável pela logística executa para pedir uma caixa para uma torre. Na figura está só representado o pedido para uma secção, se uma torre realizar mais que uma secção, o processo repete-se conforme indicado na figura.

Torre Pede Caixas aos armazéns (Resumo)

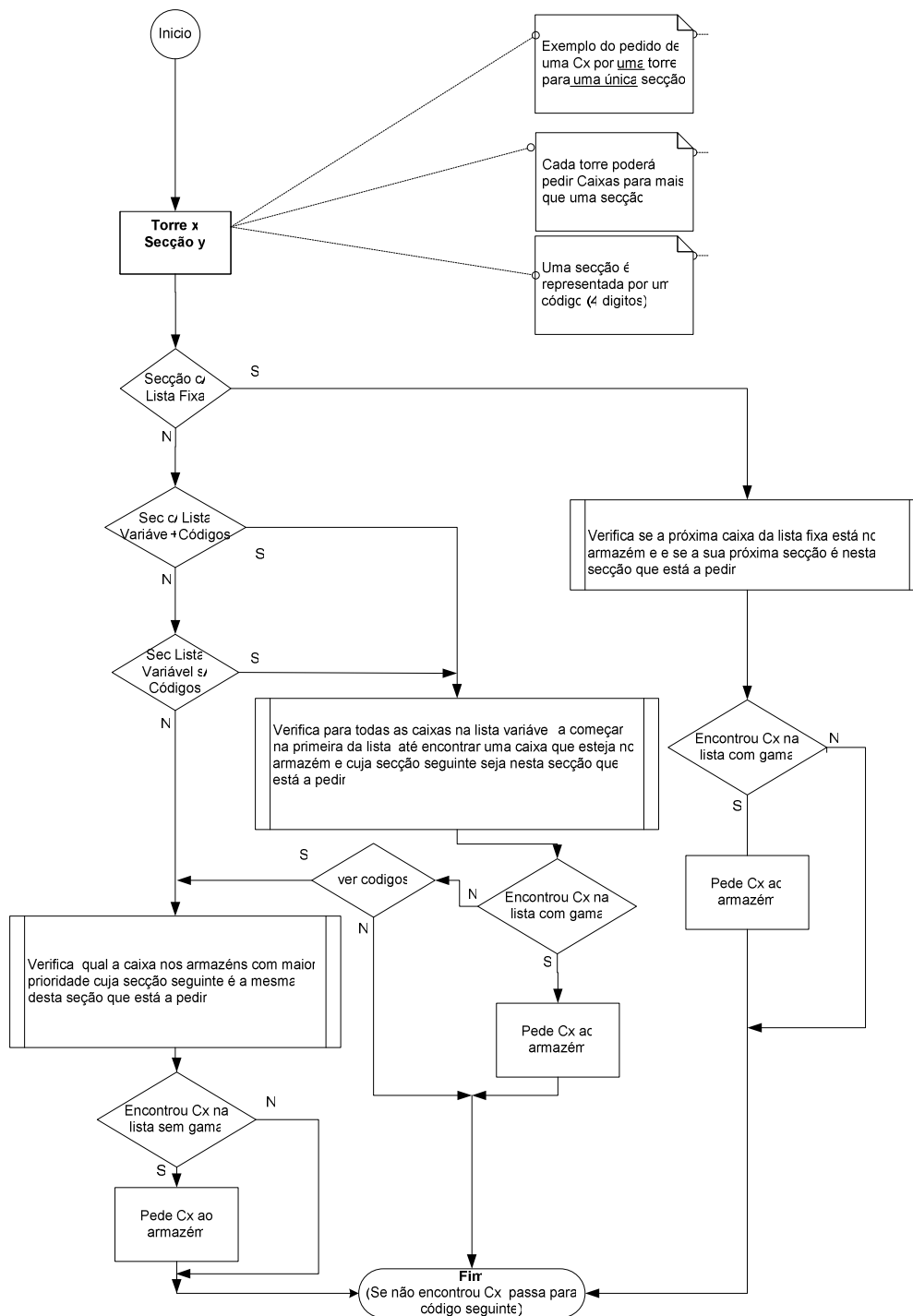


Figura 3-22: Fluxograma de uma torre a pedir uma caixa para uma secção

Quando o PLC encontra a caixa pretendida num dos armazéns, escreve-a numa lista de ‘caixas pedidas aos armazéns’. Esta lista serve de interface entre o PLC responsável pela logística interna e os PLC’s do sistema de gestão dos armazéns, ver Figura 3-23.

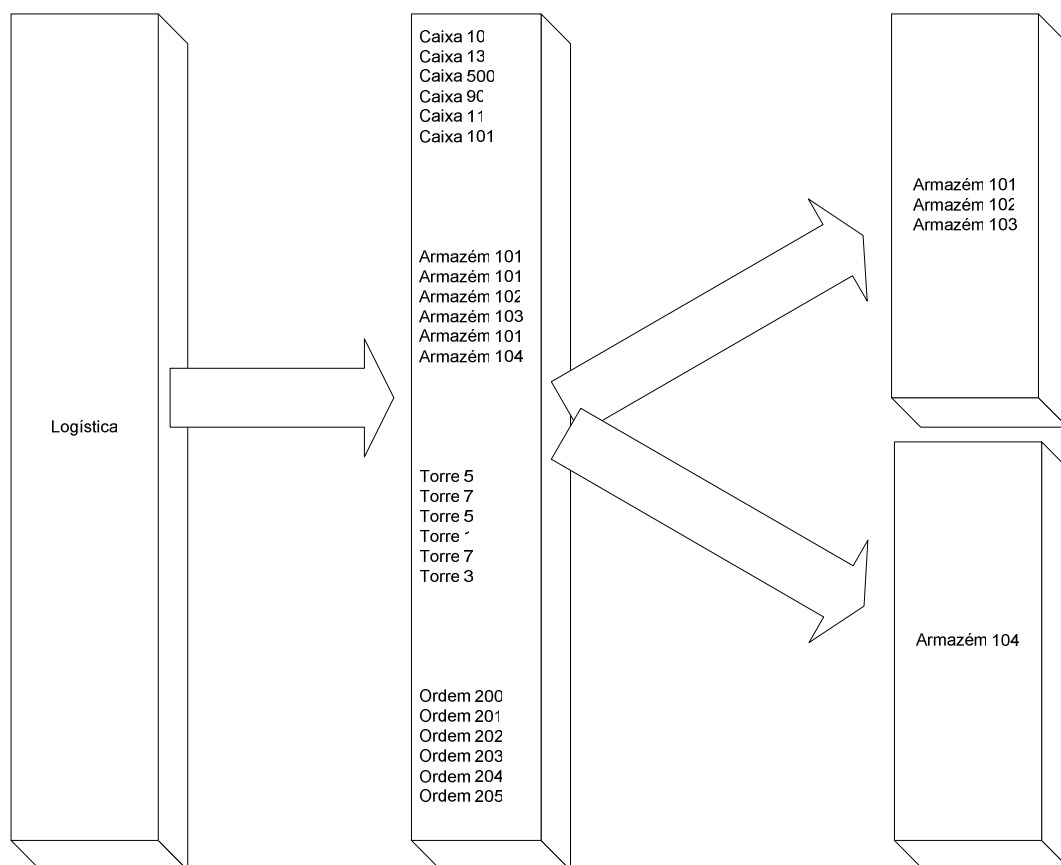


Figura 3-23: Lista de caixas pedidas pelas torres

A logística vai escrevendo na lista as caixas pretendidas pelas torres. É indicado ao sistema de armazenagem qual o nº da caixa pretendida, o armazém onde se encontra a caixa, a torre destino e o número de ordem do pedido. O sistema de armazenagem vai tentar enviar as caixas para as torres, respeitando a ordem dos pedidos.

Sempre que uma torre recebe uma caixa realiza um conjunto de procedimentos representados na Figura 3-24.

Sempre que chega uma caixa a uma torre, através do número da caixa o PLC responsável pela gestão da logística sabe qual a gama operatória e incrementa o apontador que indica qual a próxima operação (qual a próxima secção) para a operação seguinte. Verifica também se o número de caixas na secção, atingiu o máximo e caso tenha atingido o máximo, bloqueia essa secção até que o número de caixas na secção seja inferior ao máximo.

Torre recebe Cx

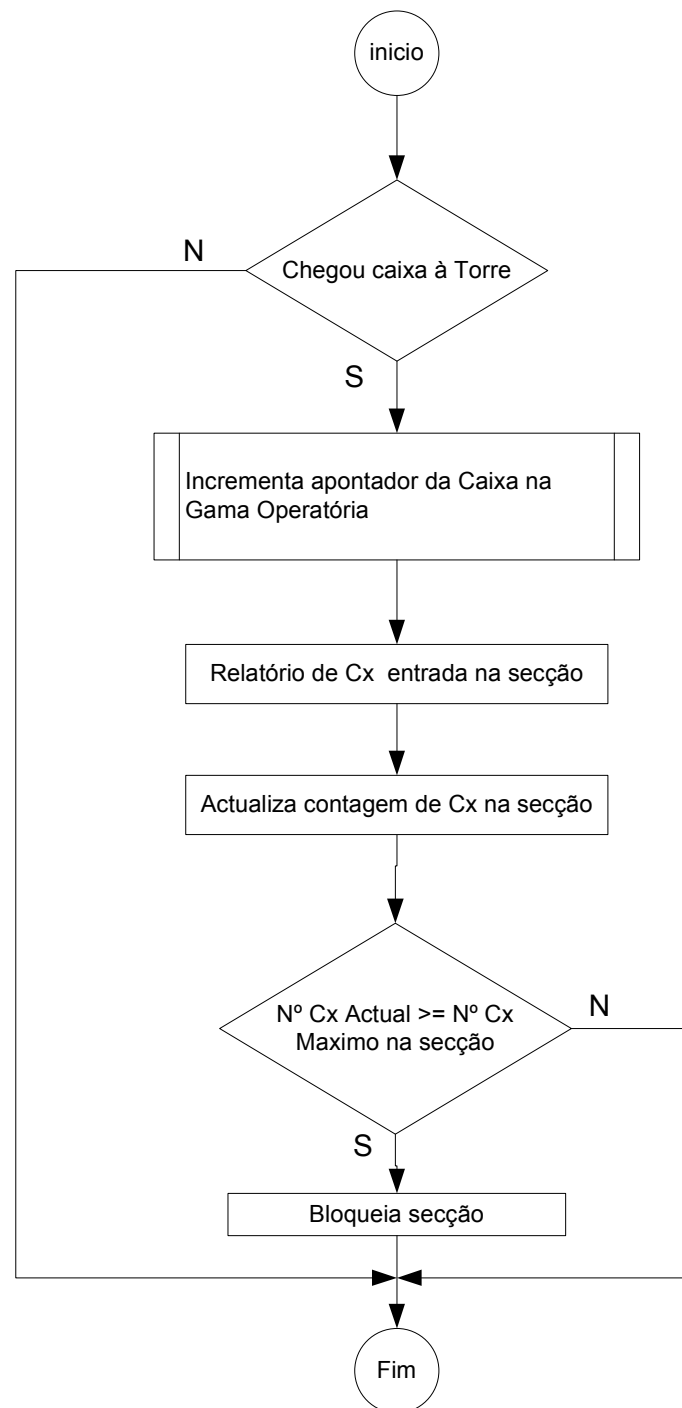


Figura 3-24: Entrada de caixa numa secção

Sempre que um contentor chega a uma secção para realizar uma operação que não é a última da gama operatória, ele após sofrer as operações necessárias na secção, terá que ser devolvido a um armazém antes de ir para outra secção realizar a operação seguinte. O operador coloca então o contentor no buffer de caixas de entrada na torre (saída da secção) e é despoletado o procedimento da Figura 3-25.

Torre Envia Cx

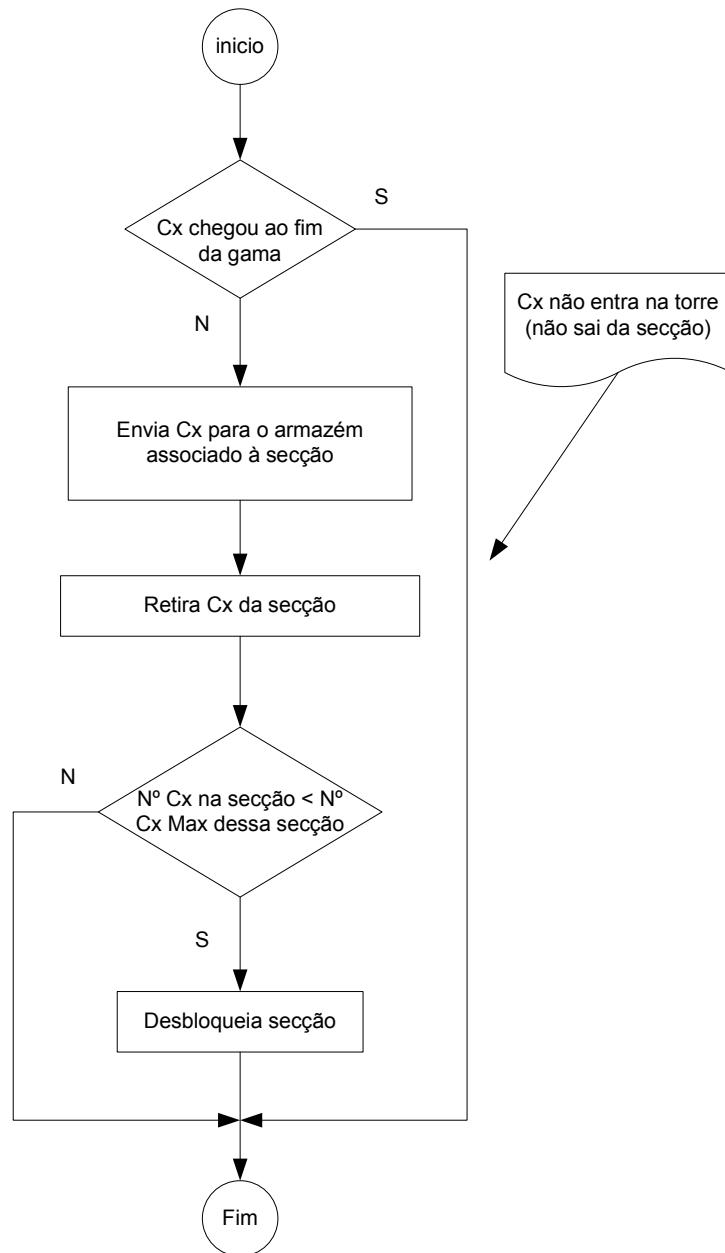


Figura 3-25: Saída de uma caixa de uma secção

4 Desenvolvimento do Modelo

4.1 Introdução

No capítulo anterior apresentou-se o sistema de movimentação e armazenamento de produto interno, instalado na empresa Codizo. Neste capítulo apresenta-se o processo de concepção e desenvolvimento do modelo de simulação. Para a implementação dos modelos em simulação, utilizou-se a ferramenta de simulação Arena, onde foi desenhado o modelo característico do sistema de logística interna na empresa Codizo. Este modelo, representação do sistema real, contém também, módulos de controlo, elaborados em VBA para aplicar os critérios e algoritmos instalados nos PLC's já descritos no capítulo anterior, que controlam o sistema real de movimentação.

Neste estudo, a simulação foi feita utilizando-se o 'software' de simulação Arena, versão 10.0, com uma cópia disponibilizada para este estudo pelo INESC Porto.

4.2 Desenvolvimento do caso de estudo

O caso de estudo apresentado no terceiro capítulo com o nome 'Sistema de Logística Interna' referente ao sistema instalado na empresa Codizo, descreveu o sistema, composto por caixas, torres, tapetes e armazéns de produto interno e algoritmos dos PLC's responsáveis pela gestão da movimentação e armazenamento das caixas.

Neste capítulo será realizada uma descrição das fases consideradas mais importantes no desenvolvimento do modelo:

- lógica utilizada entre os armazéns e a logística (tapetes);
- o processo de arranque do sistema;
- a criação da entidade caixa;
- a configuração do sistema;
- o processo de reentrada das entidades no sistema;

- a lógica utilizada nos pontos de entrada e saída das caixas das secções;
- a lógica utilizada na movimentação das caixas nos tapetes com cruzamentos e entroncamentos;
- a animação.

À medida que os módulos do Arena são apresentados no texto, será realizada uma descrição sucinta dos mesmos.

4.2.1 Lógica de interface entre os armazéns e a logística

A lógica para pedidos de caixas, utilizada na instalação pelos autómatos já foi referida no capítulo três, ver Figura 3-23. No sistema de simulação houve a preocupação de tentar utilizar a mesma lógica.

Utilizando um conjunto de funções no VBA foram implementados todos os algoritmos dos pedidos de caixas aos armazéns e o atendimento destes às secções, ver Figura 4-1. Toda a informação (caixas que estão nos armazéns, a gama operatória de todas as caixas e o estado das caixas) está armazenada em memória do PC, utilizando 'arrays'. Estas funções são chamadas sempre que acontece um evento em locais estratégicos colocados ao longo de todo o sistema de simulação, de maneira a utilizar o menor número possível os recursos do computador. Quantas mais vezes são chamadas estas funções mais lenta se torna a simulação.

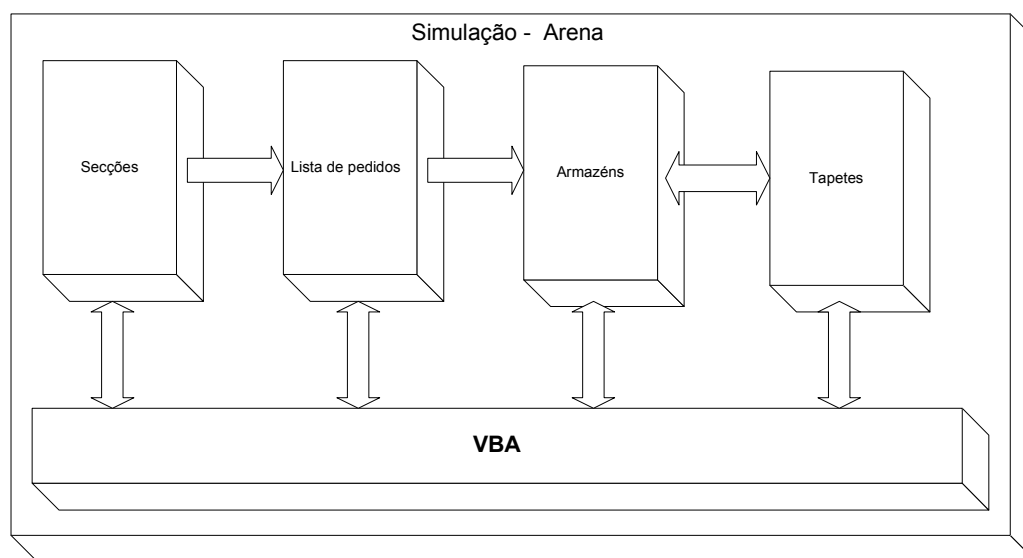


Figura 4-1: Sistema de Simulação

De seguida vão ser explicadas as funções mais importantes do sistema de simulação, desenvolvidas em VBA:

A função **‘Pedidos101’**: contém as funções necessárias para escrever na lista de pedidos, e pedir a(s) caixa(s) ao(s) armazém(ens), enviando sinais para os blocos ‘Hold’ nos respectivos armazéns. Esta função contém as seguintes funções: ‘EscreveListaPedidosArmazens’, ‘VerificaSePedidos’ e ‘EnviaSinalSaidaArmazem’. A função Pedidos101 é executada na chegada de caixas às entradas dos armazéns, na saída dos armazéns e na chegada às torres.

A função **‘EscreveListaPedidosArmazens’**: esta função escreve na lista de pedidos, uma caixa para uma secção. A função percorre todas as secções, para cada, verifica se existe nos armazéns uma caixa para a secção que está a pedir caixa. Para uma secção estar a pedir caixa é necessário que as caixas no ‘buffer’ de saída da torre (entrada de secção) onde está localizada a secção não tenham atingido o limite permitido na configuração. A lista é limitada, só escreve na lista se as condições anteriores forem satisfeitas e houver espaço na lista para escrever.

A função **‘VerificaSePedidos’**: esta função verifica se há pedidos na lista de pedidos, ver Figura 4-1. Tenta encontrar na lista de pedidos uma caixa de cada armazém com maior prioridade (menor número de ordem). Só há novos pedidos para atender se existir caixa num armazém e o último pedido a esse armazém já tiver sido atendido, por isso, para cada armazém verifica se existem ainda caixas a movimentarem-se na direcção da saída do armazém, ou seja, se o pedido anterior naquele armazém já foi atendido pelo armazém. Se houver então uma caixa que possa sair do armazém, verifica onde fisicamente está no armazém e conforme o local indica o número de caixas pretendidas para sair (se for o armazém de gravidade e a caixa pretendida estiver atrás terão que sair as quatro caixas).

A função **‘EnviaSinalSaidaArmazem’**: envia um sinal para o módulo do armazém pretendido, dando ordem para sair a(s) caixa(s). Existe um módulo ‘hold’ para cada sub-armazém nos armazéns por gravidade e um em cada alvéolo do armazém no plano horizontal.

A função **‘VerificaSeCxPodeReentrar’**: verifica se as caixas que acabaram de realizar a última operação da gama operatória, que estão no módulo ‘Hold’ com o nome ‘Recomeça’ (Figura 4-12) já podem entrar no sistema. Verifica qual o armazém destino da caixa e se esse armazém não atingiu o máximo de caixas permitido (%) é enviado um sinal para que o módulo deixe sair uma caixa na direcção do armazém. Esta função é executada na entrada e saída de todos os armazéns.

A função **‘VerificaSeCxPodeRecomecar’**: verifica se as caixas que ficaram a aguardar (porque os armazéns destinos estavam cheios) a sua vez de entrar no tapete que as levarão ao armazém destino no módulo ‘hold’ com o nome ‘Aguarda’ (Figura 4-12), já podem entrar no sistema. Verifica qual o armazém destino da caixa, e se o número de caixas nesse armazém não atingiu o máximo de caixas permitido (%), é enviado um sinal para que o módulo ‘hold’ deixe sair uma caixa na direcção do armazém. Esta função é executada na entrada e saída de todos os armazéns.

A função **‘VerificaSeTodosArmazensCheios’** verifica se os armazéns atingiram o máximo da percentagem configurada. Se o número de caixas nos armazéns é maior ou igual ao configurado o(s) armazém(ns) ficam marcados como cheios, senão ficam marcados como não cheios. Esta função é executada na entrada de todos os armazéns.

A função ‘**VerificaEntradaSeccoesNasTorres**’: verifica para cada torre se existe caixa no ‘buffer’ de entrada das secções. Se existir verifica para que secção a caixa se dirige, e verifica se essa secção atingiu o máximo de caixas configuradas para essa secção. Se não atingiu o máximo envia um sinal para o módulo ‘hold’ na torre para libertar a caixa.

4.2.2 Arranque do sistema

O arranque do sistema de simulação é realizado, utilizando três módulos, um módulo ‘create’, um módulo ‘Assign’ e dois módulos VBA, ver Figura 4-2. Tal como no sistema instalado na Codizo, as caixas entram no sistema pela torre 8, também na simulação assim acontece o mesmo.

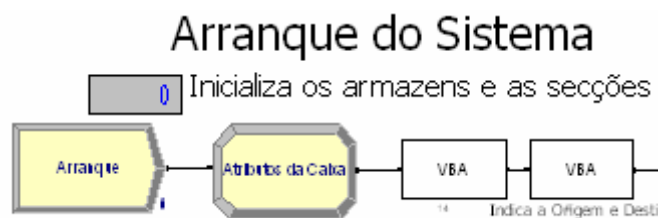


Figura 4-2: Arranque do sistema

4.2.2.1 O módulo ‘Create’:

O módulo “Create” é considerado o ponto de arranque de todo o modelo. É neste módulo que as entidades são criadas no sistema, quais quer que elas sejam, tais como por exemplo, clientes, equipamentos, caixas, carros, barcos ou aviões, qualquer coisa. As entidades, também chamadas de transacções, são tudo aquilo que “sofrerá” a acção das diversas operações lógicas do fluxo. As entidades são o “trigger” dos processos, que depois de definidas a maneira como são criadas no modelo, elas movem-se pelo sistema e fazem funcionar efectivamente os processos. No bloco ‘Create’ são inseridas várias informações importantes relativas ao modelo: o tipo da entidade, os intervalos de tempo em que são criadas, a quantidade de entidades que chegam em cada intervalo e o número máximo de entidades criadas. O tipo da entidade é um atributo da entidade, que permite identificar um determinado grupo de entidades e pode servir como parâmetro para decisões lógicas do modelo (Araújo et al. (2006)).

No módulo ‘Create’ da instalação com o nome ‘Arranque’ são criadas as **entidades caixas**. A caixa é o único tipo de entidade utilizado em todo o sistema, visto que, como foi referido anteriormente, todo o sistema gere unicamente a movimentação e armazenamento de caixas.

Figura 4-3: Criação das entidades do sistema

Com intervalos de um minuto são criadas as entidades do tipo caixa (único tipo utilizado). A quantidade de caixas a introduzir no sistema, depende da variável ‘TotalCaixasCriadas’, ver Figura 4-3.

4.2.2.2 O módulo ‘Assign’:

O ‘Arena’ permite a utilização de valores dentro das seguintes categorias: variáveis, atributos e expressões. Cada um destes recursos de armazenamento de valores tem as suas próprias características.

- As variáveis são globais, disponíveis para qualquer entidade, e os seus valores podem ser modificados ou utilizados pelas entidades
- O atributo pertence exclusivamente a uma única entidade que se move pelo sistema.
- Por meio de “expressões” podemos definir uma fórmula que é calculada pela passagem de uma entidade.

O módulo “Assign” permite ao utilizador trocar o valor de uma variável, “etiquetar” alguma entidade com um atributo específico, trocar a imagem representativa da entidade que percorre o fluxo de informações, enfim, alterar o valor de algum parâmetro ou variável do modelo (Araújo et al. (2006)).

No módulo ‘Assign’ da instalação, com o nome ‘Atributos da caixa’ (Figura 4-4) são atribuídos os **atributos** das entidades.

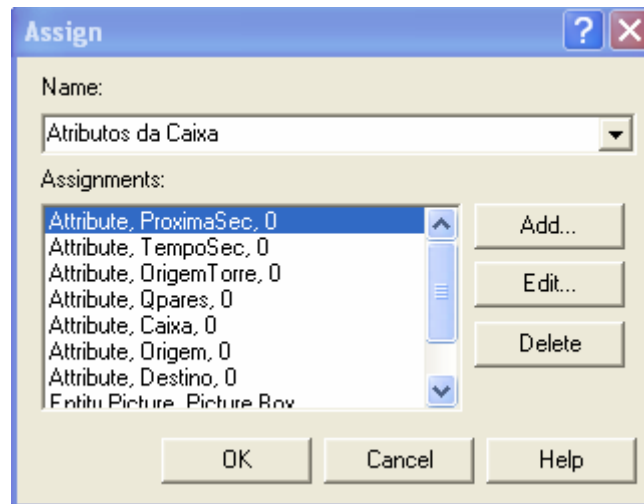


Figura 4-4: Criação dos atributos das caixas

Como já foi referido anteriormente no capítulo 2, as entidades são partes de um sistema sendo vistas como os objectos representados na simulação, descritos pelos atributos e para os quais os eventos ocorrem. Atributos são as propriedades ou características de um sistema ou de uma entidade. Neste módulo são atribuídos vários atributos a cada entidade que é criada:

- ProximaSec – Secção na gama operatória onde será realizada a operação seguinte da caixa.
- TempoSec – Tempo que a caixa permanecerá na secção seguinte.
- OrigemTorre – Torre de onde saiu a caixa – permite saber quando a caixa se desloca nos tapetes em direcção a um armazém ou uma secção, qual foi a torre de origem.
- Qpares – Quantidade de pares que a caixa transporta.
- Caixa – Número da caixa. Este número é único. No sistema real o operador identifica o número de duas maneiras: dois autocolantes com os códigos de barras, um de cada lado da caixa, e dois autocolantes com o número da caixa.
- Origem – Indica ao sistema o último cruzamento por onde a caixa passou ou torre origem.
- Destino – Indica o destino da caixa. A caixa pode ter como destino, uma torre, um armazém, um sub-armazém ou alvéolo do armazém.

4.2.2.3 O bloco ‘VBA’:

O ‘Arena’ como já referido anteriormente, possui uma interface com o Microsoft VBA, distribuído juntamente com a aplicação. O bloco VBA pertence à ‘Template’

‘Blocks’. A interface com o VBA é a partir de uma janela independente, activada de dentro de um modelo Arena.

Os atributos indicados anteriormente ficam com os valores por defeito iguais a zero. A atribuição dos valores será no módulo VBA seguinte.

No primeiro módulo VBA: neste módulo são atribuídos os valores aos atributos criados no módulo de criação dos atributos, ver Figura 4-2. Este módulo, depois dos atributos preenchidos, coloca ainda algumas caixas no seu destino, para abreviar o tempo de espera da simulação. Somente os três primeiros módulos do armazém 101 e as três secções abastecidas pela torre é que receberam as caixas, porque o utilizador da simulação tem a opção de deixar encher os armazéns e não havia necessidade de colocar mais caixas. Actualiza também os ‘arrays’ de memória

No segundo módulo VBA: neste módulo, ver Figura 4-2, é executada uma função que calcula o destino de uma caixa, tendo como parâmetro de entrada o número da caixa. Com base no destino calculado o sistema verifica se a próxima operação é realizada na torre 8 (torre de entrada de caixas no sistema), se for o caso então o sistema dá como realizada a primeira operação e volta a verificar qual o destino da operação seguinte na gama operatória da caixa, e finalmente preenche o atributo destino com o resultado do cálculo. Actualiza também os ‘arrays’ de memória

4.2.3 Criar a entidade ‘caixa’

Na simulação, uma das primeiras funções do VBA a ser executada é o seu ‘ModelLogic_RunBeginSimulation()’. Nesta função da simulação é criada a gama operatória de cada caixa. Ao longo do estudo foram realizadas três tentativas para a criação da gama operatória, (1) Leitura da gama proveniente de um ficheiro em formato Excel, (2) Leitura da gama proveniente da base de dados instalada na empresa Codizo (SQL) e (3) Criação da gama na própria simulação.

- A **primeira tentativa** consistiu na Leitura de um ficheiro Excel, ver Figura 4-5, com o intervalo de chegada entre as caixas, a Ordem de Fabrico a que pertence as caixas, o número das caixas, quantidade de pares de sapatos em cada caixa, e os locais (secções) por onde devem passar as caixas e os tempos previstos para cada caixa nesses locais.

TChegada	OrdFab	Caixas	QPares	sec1	TSec1	sec2	TSec2	Sec3	TSec3	sec4	TSec4	Sec5	TSec5
1	1	1	10	7	20	7	20	5	20	3	20	2	20
2	1	2	10	7	20	7	20	5	20	3	20	2	20
3	1	3	10	7	20	7	20	5	20	3	20	2	20
4	1	4	10	7	20	7	20	5	20	3	20	2	20
5	1	5	10	7	20	7	20	5	20	3	20	2	20
6	1	6	10	7	20	7	20	5	20	3	20	2	20
7	1	7	10	7	20	7	20	5	20	3	20	2	20
8	1	8	10	7	20	7	20	5	20	3	20	2	20
9	1	9	10	7	20	7	20	5	20	3	20	2	20

Figura 4-5: Chegada de caixas utilizando em ficheiro Excel

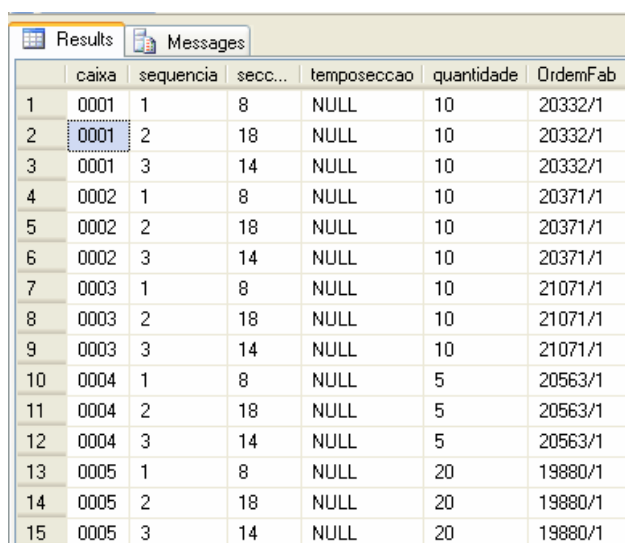
Como as caixas entram no sistema através de uma torre, e como é necessário indicar ao sistema qual o armazém que abastece uma determinada secção, então no mesmo ficheiro mas noutra folha ‘Excel’, estava indicado qual o armazém para onde deveria seguir a caixa quando entrasse na torre em direcção a um armazém, ver Figura 4-6.

Sec	Arm
1	103
2	104
3	104
4	104
5	102
6	102
7	101
8	101

Figura 4-6: Armazéns que abastecem as secções

Esta primeira tentativa de criação de entidades foi abandonada pelo facto de ser pouco flexível, ou seja, estava no ficheiro uma quantidade fixa de caixas que entrariam no sistema. Também no sistema instalado os PLC’s que gerem todo o sistema, não sabem o tempo previsto que cada caixa deve estar em cada secção, porque não lhes é fornecido.

A **segunda tentativa** foi a utilização em ‘online’ e em tempo real da base de dados do sistema instalado. O sistema no seu arranque estabelecia uma ligação à BD instalada na empresa Codizo e realizava uma leitura ‘Query’ à BD, ver Figura 4-7. O resultado da leitura retornava os dados necessários para a simulação. Nesta segunda tentativa, tinha como vantagem em relação à anterior o facto da gama operatória e a quantidade de pares em cada caixa, ser real, mas tinha como inconveniente a pouca quantidade de caixas existente no sistema nos instantes das leituras, devido ao facto das caixas só existirem no sistema informático desde o instante que são associadas ao sistema no momento da entrada para realizarem a primeira operação, até que efectuem a última operação.



	caixa	sequencia	secc...	temposeccao	quantidade	OrdemFab
1	0001	1	8	NULL	10	20332/1
2	0001	2	18	NULL	10	20332/1
3	0001	3	14	NULL	10	20332/1
4	0002	1	8	NULL	10	20371/1
5	0002	2	18	NULL	10	20371/1
6	0002	3	14	NULL	10	20371/1
7	0003	1	8	NULL	10	21071/1
8	0003	2	18	NULL	10	21071/1
9	0003	3	14	NULL	10	21071/1
10	0004	1	8	NULL	5	20563/1
11	0004	2	18	NULL	5	20563/1
12	0004	3	14	NULL	5	20563/1
13	0005	1	8	NULL	20	19880/1
14	0005	2	18	NULL	20	19880/1
15	0005	3	14	NULL	20	19880/1

Figura 4-7: ‘Query’ à Base de dados instalada na Codizo

Esta tentativa foi abandonada porque apesar de se basear em dados reais, a quantidade de caixas poderia ser pouca e além disso era pouco flexível.

A **terceira tentativa** Consistiu na criação da gama operatória utilizando o VBA. No modelo lógico do VBA foram então criadas um conjunto de gamas operatórias, por exemplo, num cenário, 7 gamas operatórias diferentes (gama 1 até gama 7), noutro cenário poderão ser outras gamas. Conforme as caixas são criadas pelo sistema de simulação, é atribuída a cada uma a sua gama operatória conforme descrito de seguida:

- Para as caixas nº 1, caixa nº 8, Caixa nº15, Caixa nº22 Até ao limite de caixas imposto pelo utilizador da simulação, foi atribuída a gama1
- Para as caixas nº 2, caixa nº 9, Caixa nº16, Caixa nº23 Até ao limite de caixas imposto pelo utilizador da simulação, foi atribuída a gama2
- Para as caixas nº 3, caixa nº 10, Caixa nº17, Caixa nº24 Até ao limite de caixas imposto pelo utilizador da simulação, foi atribuída a gama3
- Para as caixas nº 4, caixa nº 11, Caixa nº18, Caixa nº25 Até ao limite de caixas imposto pelo utilizador da simulação, foi atribuída a gama4
- Para as caixas nº 5, caixa nº 12, Caixa nº19, Caixa nº26 Até ao limite de caixas imposto pelo utilizador da simulação, foi atribuída a gama5
- Para as caixas nº 6, caixa nº 13, Caixa nº20, Caixa nº27 Até ao limite de caixas imposto pelo utilizador da simulação, foi atribuída a gama6
- Para as caixas nº 7 caixa nº 14, Caixa nº21, Caixa nº28 Até ao limite de caixas imposto pelo utilizador da simulação, foi atribuída a gama7

A informação que acompanha as gamas operatórias desde a gama 1 até à gama 7, é descrita de seguida:

- Cx – Número da Caixa
- Seq - Número da sequencia das operações (seguida sem saltos na gama operatória)
- Sec - Código da secção
- T. Op - Tempo da realização das operações na secção
- Q.Pares - Quantidade de pares na Caixa
- O.F. - Ordem de fabrico

As sete gamas operatórias (um possível cenário):

Cx	Seq	Sec	T.Op.	Q. Pares	O.F.
1	1	6	0	X1	2082/01
1	2	8	0	X1	2082/01
1	3	10	0	X1	2082/01
1	4	12	0	X1	2082/01

Tabela 4-1: Gama 1

Cx	Seq	Sec	T.Op.	Q. Pares	O.F.
9	1	6	0	X2	2082/01
9	2	8	0	X2	2082/01
9	3	10	0	X2	2082/01
9	4	18	0	X2	2082/01
9	5	14	0	X2	2082/01

Tabela 4-2: Gama 2

Cx	Seq	Sec	T.Op.	Q. Pares	O.F.
17	1	6	0	X3	2082/01
17	2	8	0	X3	2082/01
17	3	10	0	X3	2082/01
17	4	16	0	X3	2082/01

Tabela 4-3: Gama 3

Cx	Seq	Sec	T.Op.	Q. Pares	O.F.
25	1	20	0	X4	2082/01
25	2	8	0	X4	2082/01
25	3	10	0	X4	2082/01
25	4	12	0	X4	2082/01

Tabela 4-4: Gama 4

Cx	Seq	Sec	T.Op.	Q. Pares	O.F.
19	1	20	0	X5	2082/01
19	2	8	0	X5	2082/01
19	3	10	0	X5	2082/01
19	4	18	0	X5	2082/01

Tabela 4-5: Gama 5

Cx	Seq	Sec	T.Op.	Q. Pares	O.F.
20	1	20	0	X6	2082/01
20	2	8	0	X6	2082/01
20	3	10	0	X6	2082/01
20	4	18	0	X6	2082/01
20	4	14	0	X6	2082/01

Tabela 4-6: Gama 6

Cx	Seq	Sec	T.Op.	Q. Pares	O.F.
21	1	20	0	X7	2082/01
21	2	8	0	X7	2082/01
21	3	10	0	X7	2082/01
21	4	16	0	X7	2082/01

Tabela 4-7: Gama 7

- As secções de produção são abastecidas pelas torres:

Sec	Torre	Nome	Descrição
6	7	Costura	Esta torre recebe e envia caixas. Os locais de recepção e envio estão localizados em espaços diferentes.
8	5	Pré-Montagem	Esta torre recebe e envia caixas. Os locais de recepção e envio estão localizados em espaços diferentes
10	6	Moldar	Esta torre recebe e envia caixas. Os locais de recepção e envio estão localizados em espaços diferentes. O local de envio é o mesmo da Torre 5.
12	1	Montagem I	Esta torre só recebe caixas
14	2	Montagem II	Esta torre só recebe caixas
16	4	Montagem III	Esta torre só recebe caixas
18	3	Montagem2 Enformar	Esta torre encontra-se no espaço de Montagem 2. Recebe e envia caixas
20	8	Subcontratados	Esta torre recebe e envia caixas.

Tabela 4-8: Secções - Torres

- O tempo previsto para a realização das operações numa secção (T. Op.) está a zero, porque será calculado no instante de entrada nas secções.

- A quantidade de pares em cada caixa é aleatória (X1...X7), com base num cálculo realizado no instante da criação da gama operatória:

$$X1... X7 = \text{Int}((10 * \text{Rnd}) + \text{Quant})$$

Os valores a colocar em X1... X7 dependem da variável 'Quant'

$$\text{Quant} \leq X1... X7 \leq \text{Quant} + 10$$

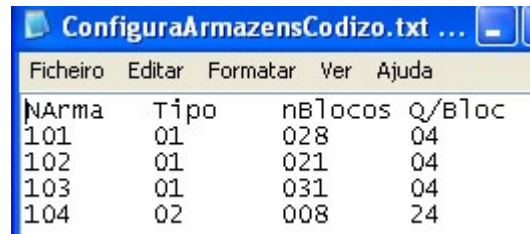
- A Ordem de Fabrico (O.F.) é sempre a mesma.

4.2.4 Configuração do sistema

No arranque do sistema de simulação, o VBA lê três ficheiros em formato de texto com a configuração do sistema.

Ficheiro de configuração dos armazéns ("ConfiguraArmazensCodizo.txt").

Neste ficheiro (Figura 4-8), tem para cada armazém, o tipo de armazém (por gravidade ou na horizontal), o número de blocos e a quantidade de caixas por cada bloco.



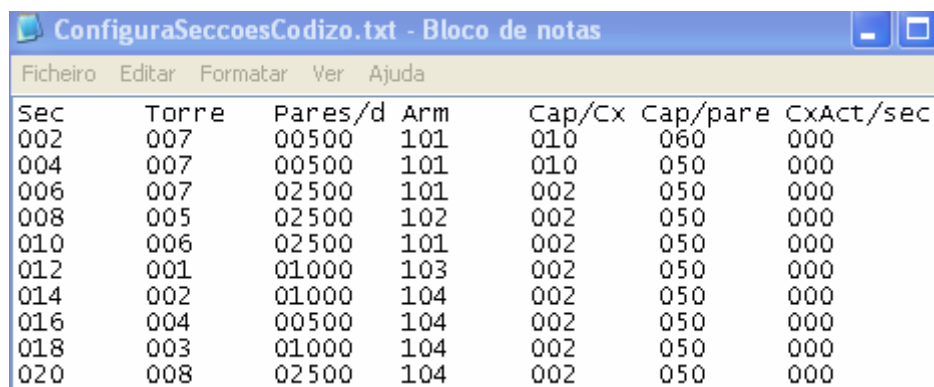
NArma	Tipo	nBlocos	Q/Bloc
101	01	028	04
102	01	021	04
103	01	031	04
104	02	008	24

Figura 4-8: Configuração de cada armazém

Este ficheiro é utilizado pelo programador da simulação. A informação contida neste ficheiro é lida no arranque da simulação e armazenada em memória ('arrays') para ser utilizada pela simulação nos diversos módulos em Visual Basic. Em termos de lógica a simulação está preparada para outros valores a colocar neste ficheiro, contudo a parte gráfica não. Deste modo, o utilizador da simulação não deve alterar o conteúdo deste ficheiro.

Ficheiro de configuração das secções ("ConfiguraSeccoesCodizo.txt").

Neste ficheiro (Figura 4-9) estão as configurações de cada secção.



Sec	Torre	Pares/d	Arm	Cap/Cx	Cap/pare	CxAct/sec
002	007	00500	101	010	060	000
004	007	00500	101	010	050	000
006	007	02500	101	002	050	000
008	005	02500	102	002	050	000
010	006	02500	101	002	050	000
012	001	01000	103	002	050	000
014	002	01000	104	002	050	000
016	004	00500	104	002	050	000
018	003	01000	104	002	050	000
020	008	02500	104	002	050	000

Figura 4-9: Configuração de cada secção

O significado de cada coluna no ficheiro de configuração das secções:

Sec - Código da secção;

Torre - Número da torre;

Pares/d - Quantidade de pares a realizar na secção;

Arm - Número do armazém que armazena a caixa antes de ir para a secção;

Cap/Cx - Quantidade de caixas em cada secção;

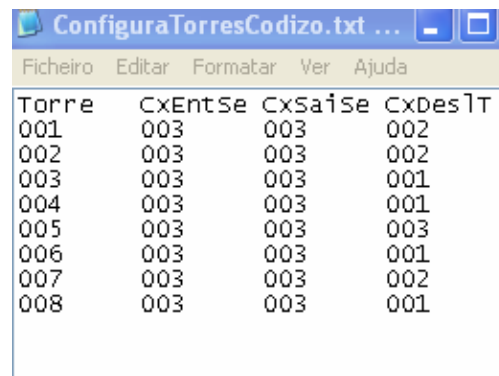
Cap/pare - Capacidade de pares por secção (não utilizado);

CxAct/sec - Quantidade de caixas presentes por secção.

Este ficheiro será utilizado para testar os vários cenários possíveis na simulação, modificando os dados nas respectivas colunas.

Ficheiro de configuração das torres ("ConfiguraTorresCodizo.txt")

Neste ficheiro (Figura 4-10), estão as configurações das torres. O 'buffer' de caixas na saída das secções existe na instalação da Codizo e foi considerado no desenvolvimento da simulação, mas acabou por ser abandonado, pelo facto de não ter nenhuma influência, ou seja, todas as caixas que acabam as operações nas secções e pretendem voltar a entrar no sistema de movimentação para se deslocarem na direcção de um armazém entram automaticamente num 'buffer' até que o sistema as possa colocar nos tapetes.



Torre	CxEntSe	CxSaiSe	CxDeslT
001	003	003	002
002	003	003	002
003	003	003	001
004	003	003	001
005	003	003	003
006	003	003	001
007	003	003	002
008	003	003	001

Figura 4-10: Configuração das Torres

O significado de cada coluna no ficheiro de configuração das torres:

Torre - Número da torre;

CxEntSe - Quantidade de caixas no 'buffer' de saída (entrada da secção) de cada torre;

CxSaiSe - Quantidade de caixas no 'buffer' de entrada (saída da secção) de cada torre;

CxDeslT - Quantidade de caixas que se podem deslocar ao mesmo tempo a partir de um armazém, na direcção de uma torre.

Écran de arranque

Neste écran (Figura 4-11) o utilizador poderá manipular a variável total de caixas a criar no sistema; activar ou desactivar as torres, permitindo que estas peçam ou não caixas aos armazéns; indicar a percentagem possível de caixas em cada armazém; o número de pares pretendidos a realizar nas secções; o número mínimo de pares em cada caixa; o número de pares (aleatório) máximo a adicionar ao número mínimo de pares em cada caixa; a capacidade de caixas nas secções e o número de caixas máximo a deslocarem-se simultaneamente na direcção das torres. A variável ‘Siman’ com o nome de “TotalCaixasCriadas” é utilizada para armazenar o número de caixas que o sistema deve criar. Esta variável no instante do arranque do sistema tem por defeito o valor 250, mas poderá ser modificada pelo utilizador da simulação conforme indicado no écran de arranque do sistema. Este écran permite também que o utilizador indique quais as torres que estão em funcionamento através da activação das ‘check box’. O écran permite ainda indicar a percentagem possível de ocupação para cada armazém.

O écran aparece também sempre que o utilizador arrancar novamente o sistema, depois de uma paragem. Desta forma, a utilização deste écran para efeitos de testes é bastante útil, porque permite por exemplo, que os armazéns sejam preenchidos, indicando ao sistema que as torres onde serão realizadas as operações das caixas estão desactivas, sempre que seja necessário testar vários cenários da simulação.

Torres Activas		Armazéns % Ocupação	
<input checked="" type="checkbox"/> Torre1		Arm101	95 %
<input checked="" type="checkbox"/> Torre2		Arm102	95 %
<input checked="" type="checkbox"/> Torre3		Arm103	95 %
<input checked="" type="checkbox"/> Torre4		Arm104	98 %
<input checked="" type="checkbox"/> Torre5			
<input checked="" type="checkbox"/> Torre6			
<input checked="" type="checkbox"/> Torre7			
<input checked="" type="checkbox"/> Torre8			

Configuração	
Nº Caixas a Criar	350
Sec6 p/d	200
Sec8 p/d	2500
Sec10 p/d	1000
Sec12 p/d	1000
Sec14 p/d	1000
Sec16 p/d	500
Sec18 p/d	1000
Sec20 p/d	2500
Media Pares	20
Delta Pares	0
CapCxSec	4
CxDesIT	2

Sair

Figura 4-11: Écran de arranque do sistema

4.2.5 Reentrada de caixas no sistema

4.2.5.1 O módulo ‘Hold’:

Este módulo permite que uma entidade (caixa) ao entrar no bloco, seja colocada numa fila de espera, para aguardar um sinal ‘Wait for signal’, esperar que uma determinada condição se torne verdadeira ‘Scan for condition’ ou ficar parada indefinidamente

(deve ser posteriormente removida com o bloco Remove). Se uma caixa aguarda um sinal, o bloco ‘Signal’ deverá ser usado para permitir que a caixa se mova para o próximo bloco (Araújo et al. (2006)).

A caixa permanecerá no bloco a aguardar que uma condição se torne verdadeira (seja numa fila interna ou numa fila especificada) até que a(s) condição(ões) se torne(m) verdadeira(s). Se a caixa permanecer parada por um tempo indeterminado ‘Infinite hold’, o bloco Remove é usado para permitir que ela siga o seu processamento nos blocos seguintes (Araújo et al. (2006)).

Quando uma caixa entra num bloco configurado como ‘Scan for Condition’ e não há nenhuma fila no bloco, a condição é avaliada. Se verdadeira, a entidade continua o seu caminho ao longo do modelo. Se falsa, ela é colocada na fila. Quando uma caixa chega ao bloco e já existem outras à espera na fila, a caixa é colocada de acordo com as condições de “arranque” da mesma, a primeira a entrar será a primeira a sair (FIFO) para filas internas (Araújo et al. (2006)).

Condições com expressões numéricas são avaliadas na base do verdadeiro/falso. Se o resultado da condição for zero, retorna falso e qualquer valor diferente de zero retorna verdadeiro. Se a condição está baseada num atributo, a condição muda para cada caixa que chega (Araújo et al. (2006)).

No caso do bloco, estar configurado no modo ‘Wait for Signal’, cada caixa que entre no bloco recebe um código de sinal baseado em um valor específico. Através do uso de expressões com variáveis, as caixas de um mesmo bloco podem esperar por sinais diferentes. Um bloco ‘Signal’ pode ser usado para controlar o número de entidades que serão libertadas de uma fila de espera quando ele enviar o sinal (Araújo et al. (2006)).

No sistema instalado, as caixas entram no sistema através da torre 8, e podem entrar de 2 maneiras:

- 1 - Inicialmente no instante da criação das entidades (caixas);
- 2- Depois de terem realizado a última operação voltam a entrar mas neste caso com a gama operatória criada inicialmente no ponto 1.

O sistema de simulação verifica se os armazéns estão cheios no instante de entrada de caixas no sistema, e caso estejam cheios ou quase cheios (80% da sua capacidade) obriga as caixas a esperarem no módulo ‘aguarda’, ver Figura 4-12.

Em locais estratégicos da simulação o sistema verifica para a primeira caixa da fila do módulo ‘aguarda’ qual o armazém destino, e caso o armazém esteja abaixo da capacidade definida será dada ordem para a caixa que está à frente da fila avançar.

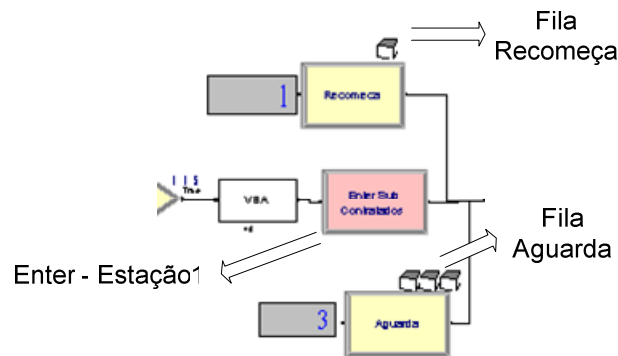


Figura 4-12: Entrada no sistema

O módulo VBA da Figura 4-12 assegura que as caixas vão para a fila do módulo Hold com o nome ‘Aguarda’ no caso do destino delas ser um dos armazéns cheios. Coloca ainda as caixas provenientes do arranque do sistema, na estação1 do sistema localizada na Torre 8. Actualiza também ‘arrays’ de memória

As caixas são colocadas na fila ‘Recomeça’ (Figura 4-12), sempre que realizam a última operação da gama operatória, esta realização é feita nas saídas das secções. O sistema ao verificar que a caixa realizou a última operação da caixa, coloca-a no módulo ‘Hold’ com o nome Recomeça à espera de um sinal para avançar.

As caixas provenientes da geração de caixas e cujo destino é a secção que está na torre 8 passa por um dos processos, ver Figura 4-13.

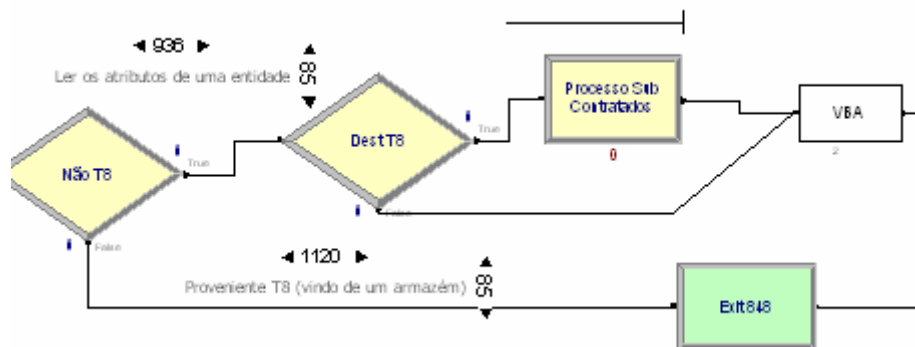


Figura 4-13: Processo Subcontratados (Proveniente da criação caixas)

4.2.5.2 O módulo ‘Process’:

Este módulo pode ser usado quando uma entidade deve executar alguma acção envolvendo um intervalo de tempo e /ou recursos, por exemplo uma caixa que sofre uma operação.

Vamos usar como exemplo a entidade “caixa” seguindo pelo fluxo do sistema. Ela deve num dado instante sofrer um processo (conjunto de operações na secção). Quando estiver na secção, obviamente outra caixa que estiver vindo “atrás” dela terá

que esperar numa fila antes de ser processada. Para isso devemos seleccionar a acção lógica “Seize Delay Release”. De forma que a caixa que entra, reserva ‘seize’ a secção para si, sofre as operações, representado por um certo atraso no seu prosseguimento no fluxo ‘delay’ e depois liberta ‘release’ a secção para que a próxima caixa possa ser processada (Araújo et al. (2006)).

Outra situação hipotética, é uma caixa que precisa sofrer três processos consecutivos na mesma secção, de forma que outra caixa não possa ser processada antes que a última tenha passado por toda a sequência. Para isso usaremos as outras opções de acção lógica. No primeiro dos três processos, seleccionamos então “Seize Delay” para que a secção seja reservada e utilizada, mas que quando a caixa deixe a secção para o próximo passo, ela não fique livre para que outra caixa entre. No segundo processo, um simples “Delay” resolve a modelação e no terceiro utilizaremos, portanto, “Delay Release” e a sequência de processos estará então libertada para o uso de uma próxima caixa (Araújo et al. (2006)).

Na utilização do módulo “process” é importante lembrar da diferença entre os conceitos de capacidade e quantidade. Ao determinar que determinado processo precisa de uma quantidade maior do que 1 de um determinado recurso significa que o processo esperará até que uma quantidade determinada do recurso esteja disponível, ou seja, um. Assim, um processo que precisa de dois funcionários para executar, pode ser modelado como precisando da quantidade 2 do recurso operadores, neste caso o modelo deve comportar uma capacidade maior que 2 para o recurso operador. Também existe a opção de modelar o recurso operador como um conjunto (set) de operadores de forma que será possível reunir estatísticas individuais para cada um dos operadores. No caso em que existam, por exemplo, três operadores e sejam necessários dois para executar um determinado processo, o Arena escolherá dois deles dentro do grupo e reservará para o processo (Araújo et al. (2006)).

Neste caso o tempo de execução é nulo (Figura 4-14), porque representa uma operação realizada pelos subcontratados, e como tal se chegou à torre 8 é porque já foi realizada a operação.

The image shows a screenshot of the 'Process' dialog box in the Arena simulation software. The dialog box has a title bar with a question mark and a close button. It contains several sections: 'Name' and 'Type' at the top, 'Logic' in the middle, and 'Delay Type', 'Units', and 'Allocation' at the bottom. The 'Name' field is set to 'Processo Sub Contratados' and 'Type' is 'Standard'. Under 'Logic', the 'Action' is 'Seize Delay Release' and 'Priority' is 'Medium(2)'. The 'Resources' list contains 'Resource, Resource T8_Sec20, 1' and '<End of list>'. Below the resources list are buttons for 'Add...', 'Edit...', and 'Delete'. At the bottom, 'Delay Type' is 'Expression', 'Units' is 'Minutes', and 'Allocation' is 'Value Added'. The 'Expression' field contains '0'. There is a checkbox for 'Report Statistics' which is checked. At the very bottom are buttons for 'OK', 'Cancel', and 'Help'.

Figura 4-14: Processo de SubContratados proveniente do arranque

4.2.6 Lógica das torres

A lógica utilizada (módulos) em cada uma das torres é idêntica. Existem dois tipos de torres, as torres que recebem e devolvem as caixas aos armazéns (Figura 4-15) como a Torre 8, e as torres que só recebem caixas dos armazéns, abastecendo as secções, ver Figura 4-16 referente à Torre1.

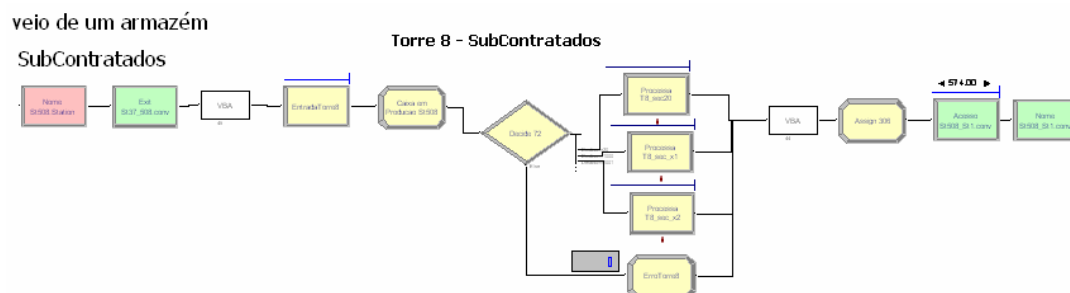


Figura 4-15: Lógica da Torre 8

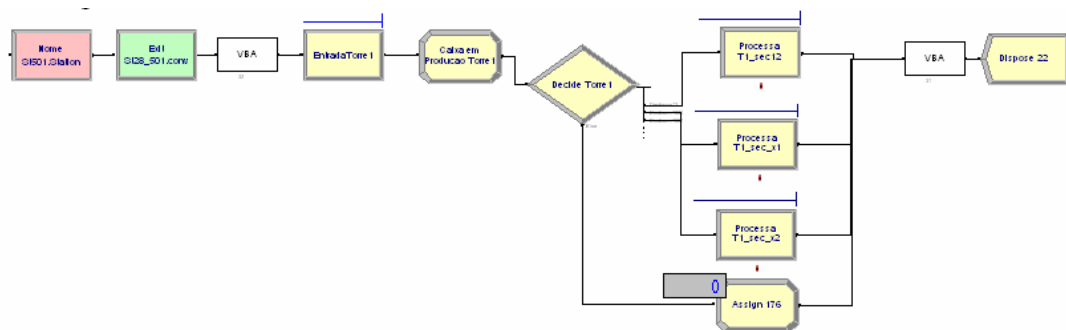


Figura 4-16: Lógica da T 1

A diferença entre os dois tipos de torres é essencialmente na lógica interna dos dois blocos em, existir no caso das torres que devolvem as caixas aos armazéns dois módulos adicionais, o módulo ‘Access’ (Figura 4-17) e o módulo ‘Convey’, ver Figura 4-18.

4.2.6.1 O módulo ‘Access’:

O módulo “Access” aloca uma ou mais células de um tapete a uma caixa para a deslocação de uma estação para outra. Após a posse das células pela entidade caixa, esta pode ser conduzida para a próxima estação. Ao chegar ao bloco, a caixa aguardará na fila, até que o número apropriado de células contíguas do tapete, estejam vazias e alinhadas com a estação (Araújo et al. (2006)).

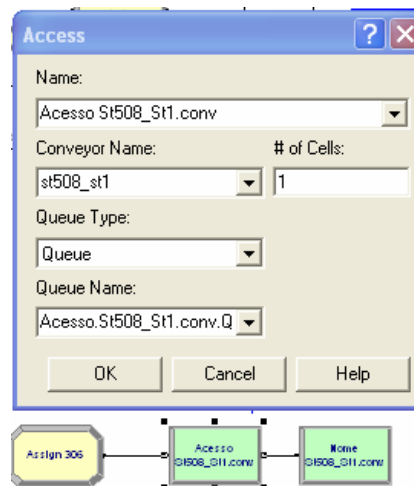


Figura 4-17: Módulo Access

O atributo 'Entity.Station' da caixa deve estar com um valor válido quando a caixa tentar aceder ao tapete de acesso aos armazéns. O número de células que deverão estar disponíveis para que a entidade aceda ao tapete dependerá do valor do atributo a colocar em '# of Cells' de cada uma das entidades caixas. No exemplo apresentado correspondente à lógica da T8, das (Figura 4-15 e Figura 4-16) o número de células ocupadas para cada caixa é 1. No módulo 'access' é necessário ainda indicar qual o nome do tapete que a caixa irá aceder, e o nome da fila de espera e a indicação do tipo de fila, ver Figura 4-17.

4.2.6.2 O módulo 'Convey':

O módulo "Convey" (Figura 4-18) é responsável pelo transporte de uma entidade caixa de uma estação para outra através de um tapete. O tempo gasto no transporte da caixa de uma estação à próxima é baseado na velocidade do tapete, especificada no bloco 'Conveyor' (Figura 4-19), e na distância entre as estações, especificada no bloco 'Segment' (Figura 4-55). O bloco 'Conveyor' e 'Segment' serão abordados mais à frente.

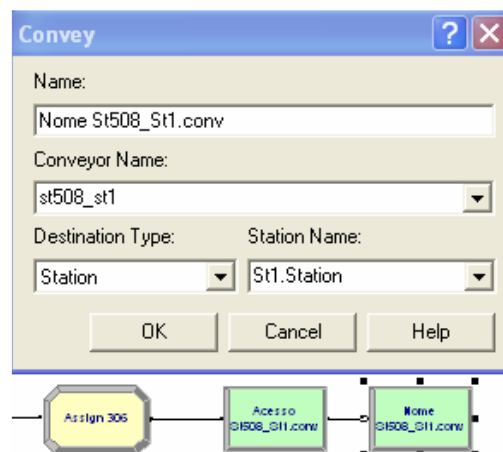


Figura 4-18: Módulo 'Convey'

Quando uma entidade caixa entra neste bloco, seu atributo ‘Entity.Station’ é definido como sendo a estação de destino. É preciso que a entidade tenha acedido às células no tapete, através do bloco ‘Access’, antes de ser conduzida à sua estação de destino ou, por outras palavras, antes de entrar ao bloco ‘Convey’. Se o nome do tapete a ser utilizado não for especificado, fica subentendido que a caixa deve ser transportada através do último tapete que ela tenha acedido (através do bloco ‘Access’).

Conveyor - Advanced Transfer										
	Name	Segment Name	Type	Velocity	Units	Cell Size	Max Cells Occupied	Accumulation Length	Initial Status	Report Statistics
323	st508_st1	st508_st1_seg	Accumulating	1.	Per Second	1	1	1	Active	<input checked="" type="checkbox"/>

Figura 4-19: Módulo 'Convey' Dados

4.2.6.3 O módulo ‘Station’:

Na lógica de todas as torres (Figura 4-15) o primeiro dos módulos é o ‘station’ (Figura 4-20). Este módulo é utilizado para informar que a caixa ocupou uma estação. As estações são utilizadas especialmente para isolar partes do modelo que funcionem independentemente de outras quer seja lógica ou fisicamente. Com a utilização da animação, é importante a utilização do módulo “station”. Pode dizer-se mesmo que são utilizadas para fins de “animação avançada” do sistema. A ideia é que todos os blocos lógicos que não envolvam movimentação física da entidade (correspondente com o sistema real) sejam agrupados numa estação. Nos casos em que a entidade se desloque de uma estação para outra é necessário que utilize um dos tipos de transporte que o Arena dispõe (Araújo et al. (2006)).

Por exemplo, no caso da torre 5, uma caixa chega a uma secção, através da torre, para ser processada, lá ela sofrerá várias acções (passará por vários postos de trabalho) mas fisicamente não estará a movimentar-se, e só quando tiver terminado todo o processamento a mesma torre a colocará num tapete que a levará para um armazém. Nesse caso, todos os blocos utilizados para representar as acções executadas na secção podem ser agrupados numa estação, e o transporte entre esta estação e o armazém, será feito por um tapete. Assim, sempre que uma entidade chega a uma estação ela é registada nesta estação e assim permanece até que passe por outra estação.

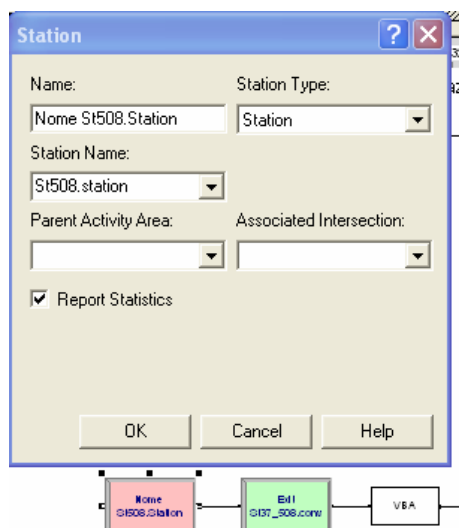


Figura 4-20: Módulo 'Station'

O módulo 'station' é bastante utilizado neste sistema de simulação porque o modelo tem um conjunto variado de tapetes utilizados para levar e trazer caixas dos armazéns para as torres e vice-versa, e necessariamente existem bastantes cruzamentos. Para indicar ao sistema que um cruzamento foi ocupado, foi utilizado o módulo 'station'. Mais à frente será explicado mais em pormenor mais à frente na secção 4.2.7

4.2.6.4 O módulo 'Exit':

O bloco 'Exit' (Figura 4-21) liberta as células ocupadas pela caixa num tapete. Uma caixa só poderá aceder a um tapete se a caixa que anteriormente esteve nesse tapete for libertada. É necessário que uma caixa tenha acedido previamente a um tapete utilizando um módulo "Access" para que possa sair do tapete através do módulo "Exit".

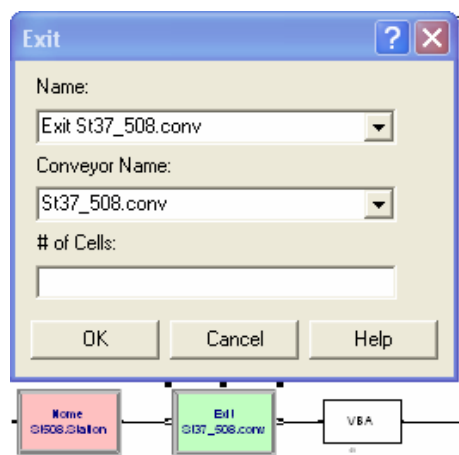


Figura 4-21: Módulo 'Exit'

Se o nome do tapete não for especificado, a caixa sairá daquele que tiver sido acedido mais recente pela mesma. Uma caixa que tenha acedido a células num tapete não precisa necessariamente ser conduzida por ele, porque é possível ela simplesmente aceder às células, sofrer um atraso e depois liberar as mesmas (Araújo et al. (2006)).

No caso do sistema de simulação e em particular numa torre (Figura 4-15), quando uma caixa chega a uma estação é libertado o tapete que a transportou (Figura 4-21). Na figura está indicado qual o tapete libertado, mas não é necessário, porque afinal a entidade simplesmente liberta aquilo que estava usando e não há necessidade de especificar-se o tapete, nem mesmo o número de células, pois serão aquelas que estão a ser ocupadas pela entidade caixa (a não ser algum caso muito específico)

O bloco VBA na entrada das secções:

O módulo VBA na entrada das secções (Figura 4-22) tem a lógica necessária para:

- Verificar se existem novos pedidos ao armazém;
- Escrever na lista de pedidos;
- Enviar sinal de saída para os vários armazéns;
- Verifica qual o destino da caixa (secção). Não esquecer que cada torre pode conter mais que uma secção. Na simulação estão previstas três secções;
- Actualiza o atributo da caixa referente ao tempo de permanência na secção (TempoSec);
- Actualiza o número de pares na secção;
- Actualiza 'arrays' de memória.

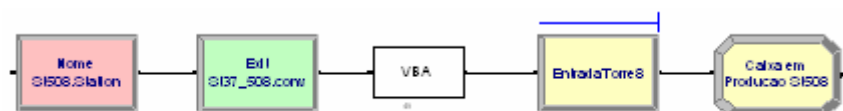


Figura 4-22: Módulo VBA entrada das Secções

O módulo 'Hold' à frente do módulo VBA é utilizado para reter as caixas à saída da torre e à entrada das secções. Este módulo representa no sistema instalado o 'buffer' de caixas na entrada das secções. O número de caixas permitido neste 'buffer' é configurado, conforme já foi referido no capítulo anterior (Figura 4-10). Este módulo (Figura 4-23), espera um sinal que faça avançar a caixa para o módulo seguinte,

retirando-a do buffer e permitindo ao sistema que despolete um novo pedido de uma caixa aos armazéns para a respectiva torre.

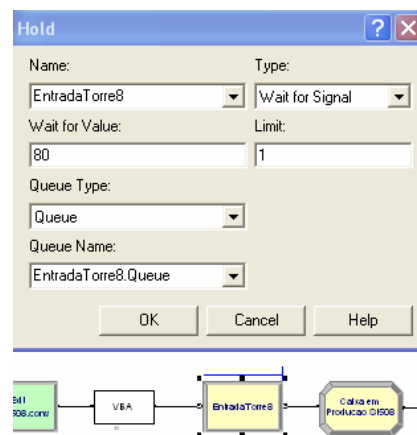


Figura 4-23: 'Buffer' entrada nas secções

O módulo ‘assign’ seguinte é utilizado para incrementar uma variável ‘siman’ que indica o número de caixas que chegaram à torre.

4.2.6.5 O módulo ‘Decide’:

O módulo “Decide” permite a tomada de decisão no fluxo das entidades no sistema. O módulo permite decisões do tipo binárias (verdadeira ou falsa), ou do tipo múltipla. O utilizador pode optar por uma escolha através de probabilidades (por exemplo: há 55% de probabilidade da caixa ir para a fila 1 e 45% de ela ir para a fila 2), de atributos (por exemplo: caixas com um determinado destino seguem para secção x e com outro destino seguem para secção y), de expressões (por exemplo: avalia-se o tamanho da fila do processo na secção x, se ela for maior que dez, a entidade segue para o processo na secção y), ou variáveis do processo.

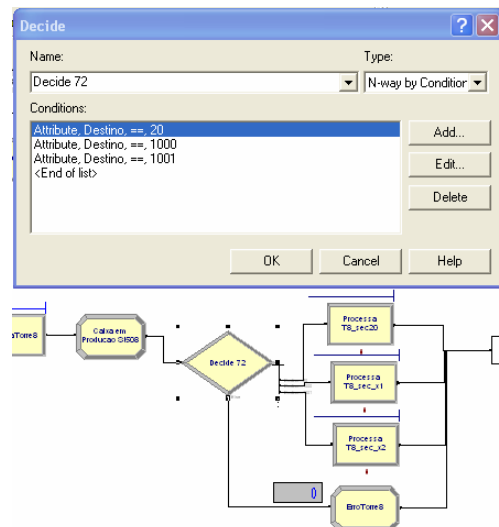


Figura 4-24: Módulo 'Decide'

Neste módulo é tomada uma decisão conforme o atributo 'destino' verificado no módulo anterior (Figura 4-22) em VBA e escrito no atributo 'destino' da caixa. No exemplo da Figura 4-24 referente à torre 8, que no sistema instalado só abastece a secção com o código 20. Os outros dois códigos 1000 e 1001 não existem, por isso as caixas vão necessariamente para a secção 20. Contudo se por algum motivo chegar alguma caixa à torre 8 cujo código da secção não seja 20 esta caixa será direccionada para um módulo 'assign' que incrementará uma variável de erro referente à torre 8. Desta maneira será possível contabilizar se o sistema de simulação enviou alguma caixa erradamente para a torre 8.

Os três módulos 'process' (Figura 4-24) seguintes nas torres são muito idênticos (Figura 4-25). No caso da torre 8 que só abastece a secção com código 20, só será utilizado um módulo 'process', mas se na instalação quiserem utilizar 3 secções na torre 8 o sistema de simulação está preparado (tem 3 módulos 'process').

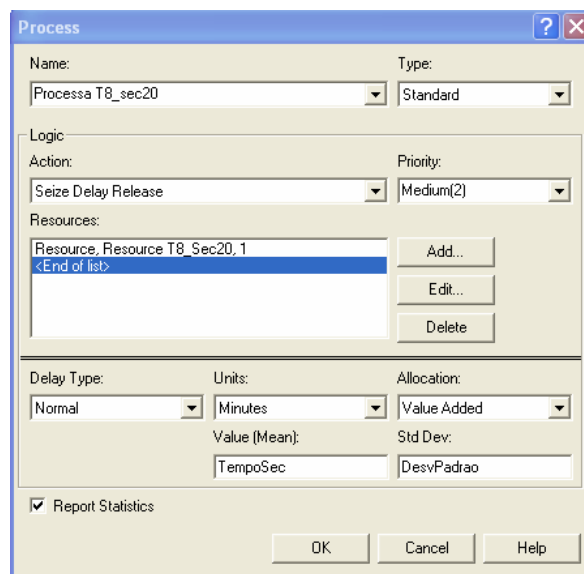


Figura 4-25: Módulo 'process' nas torres

No exemplo o tempo de atraso segue uma distribuição normal com média calculada no módulo VBA anterior (Figura 4-22) e com desvio padrão configurável pela variável “DesvPadrao”. Todos os postos de trabalho numa secção, são representados por um único recurso, conforme a Figura 4-25.

O uso de uma distribuição despromovida de comprovação, pode implicar a obtenção de resultados de simulação bastante diferentes dos reais. Na Tabela 4-9: Distribuições no Arena (Prado(2004)) segue um pequeno resumo sobre as distribuições de frequência possíveis de utilizar no Arena (Prado (2004)):

Distribuição	Abrev.	Parâmetros	Melhor aplicação
Poisson	POIS	Média	Chegada
Exponencial	EXPO	Média	Chegada
Triangular	TRIA	Min/Média/Max	Atendimento (aproximação inicial)
Uniforme	UNIF	Min/Média/Max	Atendimento (aproximação inicial)
Normal	NORM	Média/Desvio	Atendimento (tempos de máquina)
Johnson	JOHN	G,D,L,X	Atendimento
Log Neperiano	LOGN	Média Logarítmica	Atendimento
Weibul	WEIB	Beta, Alfa	Atendimento (tempo de vida dos equipamentos)
Discreta	DISC	P1,V1, ...	Chegada / Atendimento
Contínua	CONT	P1,V1, ...	Chegada / Atendimento
Erlang	ERLA	Média / K	Atendimento
Gamma	GAMM	Beta, Alfa	Atendimento (tempos de manutenção)

Tabela 4-9: Distribuições no Arena (Prado(2004))

O bloco VBA na saída das secções:

O módulo VBA na saída das secções (Figura 4-26) tem a lógica necessária para:

- Verificar o próximo destino da caixa;
- Decrementar o número de caixas na secção onde saiu a caixa;
- Incrementar o índice (apontador da próxima secção);
- Se a caixa ainda tem que fazer mais operações da gama operatória para realizar deve voltar para um armazém;
- Actualiza atributo da caixa (armazém destino);
- Actualiza atributo da caixa (nº da torre);
- Actualiza atributo da caixa (próxima secção);
- Actualiza atributo da caixa (tempo previsto);

- Se a caixa realizou a última operação da gama operatória, coloca a caixa na estação 1 (Torre 8) e insere a caixa na fila ‘Recomeça’;
- Verifica o próximo destino da caixa;
- Se o próximo destino é a torre 8, incrementar o índice (apontador da próxima secção).

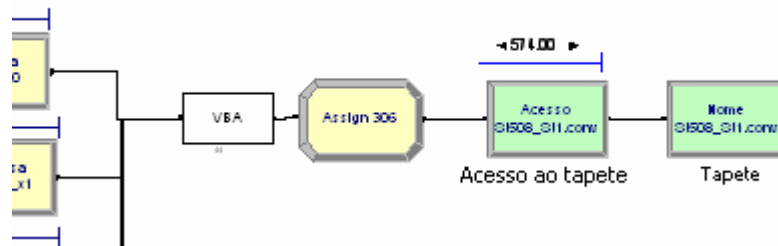


Figura 4-26: Módulos saída das torres

O módulo seguinte ‘Assign’ só modifica o atributo ‘Origem’ indicado conforme o caso, qual a torre origem a utilizar na lógica dos tapetes a ver mais à frente em “Lógica dos tapetes”.

O módulo seguinte ‘Access’ só existe nas torres que devolvem as caixas aos armazéns (Figura 4-27). Neste módulo é necessário indicar qual o transportador que a caixa vai aceder, a quantidade de células a ocupar e o nome da fila a espera em caso de esta não poder aceder ao tapete pretendido.

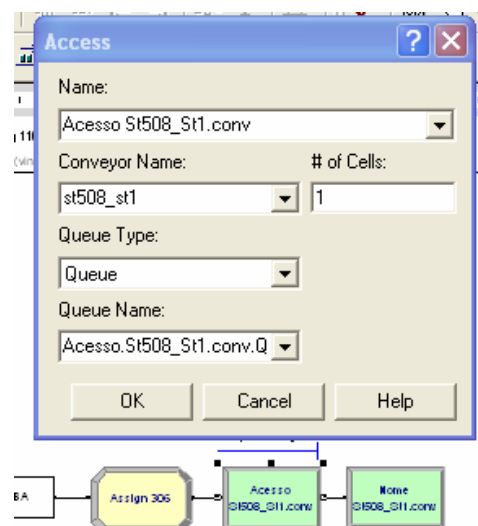


Figura 4-27: Módulo Access saída de uma torre

Por fim o módulo ‘convey’ que também só existe na lógica das torres que devolvem caixas aos armazéns, ver Figura 4-28. Este módulo indica ao sistema qual o tapete que irá devolver a caixa ao armazém e qual a estação destino do tapete.

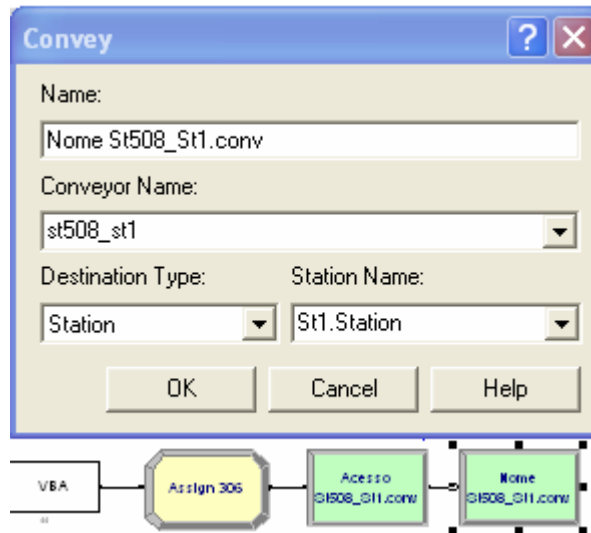


Figura 4-28: Módulo Convey na saída de uma torre

4.2.7 Lógica dos tapetes

Os tapetes são o meio de transporte utilizado para deslocar as caixas das torres para os armazéns e dos armazéns para as torres.

Devido à complexidade de tapetes e armazéns existente na instalação (Figura 3-15), com vários entroncamentos, a lógica utilizada para gerir os armazéns é descrita de seguida recorrendo a exemplos típicos de cruzamentos (entroncamentos).

Os transportadores fixos são um dos dispositivos utilizados para movimentar uma entidade caixa de uma estação para outra. Denominam-se por fixos, por não se movimentarem em conjunto com a caixa, apenas fazendo com que esta avance.

Um exemplo típico deste caso, são as telas transportadoras. Esses dispositivos são definidos pelos pares de estações que eles ligam e pela distância entre elas, denominada segmento.

As caixas devem ser colocadas sobre o transportador fixo em qualquer uma dessas estações para poderem então ser transportadas até à estação destino. Cada tapete é dividido em várias células que representam, cada uma, a menor unidade de espaço requerido para conter uma caixa apenas. Quando uma caixa entra no transportador, ela ocupa uma ou mais células, dependendo de seu tamanho, e impede que outras caixas entrem no transportador até que ela se tenha movimentado e novos espaços apareçam livres (Araújo et al. (2006)).

Existem dois tipos de tapetes, os tapetes de acumulação ‘Accumulating Conveyors’ e os não-acumulação ‘Non-Accumulating’. Quando uma caixa pára o seu movimento, as outras caixas podem parar, ou não, o efeito sobre as outras caixas que também estão em movimento depende do tipo de transportador utilizado. Os tapetes acumulativos bloqueiam o avanço das caixas que se aproximam daquela paragem,

enquanto as outras caixas continuam normalmente a movimentar-se. Já no caso dos tapetes não-acumulativos ('Non-Accumulating Conveyors'), a movimentação de todas as caixas sobre eles, é interrompida, todas ao mesmo tempo, sempre que uma tem necessidade de parar.

4.2.7.1 Ligações entre tapetes - Logística

As ligações entre tapetes deram origem a vários tipos de ligações, conforme o sentido de deslocamento destes, cruzamento tipo 1E1S, 2E1, 1E2, nE1S, nE2S e Entroncamento.

Ligação do tipo 1E1S: Este tipo de ligação (Figura 4-29) indica que chega uma caixa por um único caminho e sai uma caixa por um único caminho. Esta ligação entre tapetes é considerado na simulação como se não houvesse nenhuma ligação, ou seja, a simulação vê tudo como um único tapete.

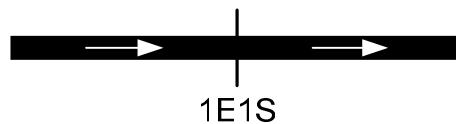


Figura 4-29: Ligação tipo 1E1S

Ligação do tipo 2E1S: Este tipo de ligação (Figura 4-30) indica que poderá chegar à ligação duas caixas ao mesmo tempo, uma por cada caminho, e a saída é por um único caminho possível.

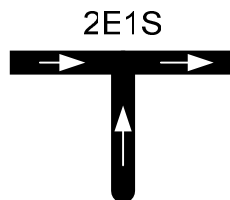


Figura 4-30: Ligação tipo 2E1S

Existem inúmeros casos destes no sistema de simulação. Todas as ligações na simulação são identificadas através de um número. Neste tipo de ligação (Figura 4-31), o sistema considera que tem:

- Uma estação;
- Um acesso ao tapete de saída;
- Uma Decisão;
- Um grupo de 2 saídas de tapete de entrada no cruzamento;

- Uma Atribuição de uma nova origem (número do cruzamento);
- Um tapete.

De seguida são descritos mais em pormenor os módulos utilizados na junção e a sua utilidade.

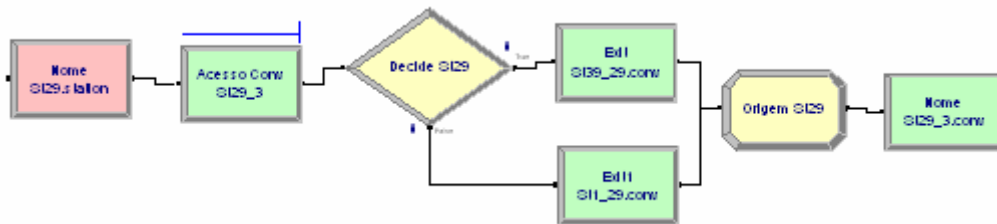


Figura 4-31: Simulação c/ ligação tipo 2E1S

No módulo referente à 'estação' é indicado nome da estação (no nome consta o número da estação, por exemplo no caso da Figura 4-31 refere-se à estação número 29.

No módulo 'acesso' refere-se o acesso ao tapete que retirará a caixa do cruzamento. Neste caso o tapete fará a ligação entre a estação 29 e a estação 3, por isso no nome do tapete de acesso e na fila, constam os números 29 e 3.

No módulo 'decide' (Figura 4-32) o sistema pergunta, qual a proveniência das caixas, dentro das duas possíveis. Conforme a resposta, a caixa vai para o módulo 'exit' respectivo. Neste tipo de cruzamento a caixa acede a um único tapete possível e sai de um dos dois tapetes possíveis, por isso existe um único módulo 'access' e dois módulos 'exit'.

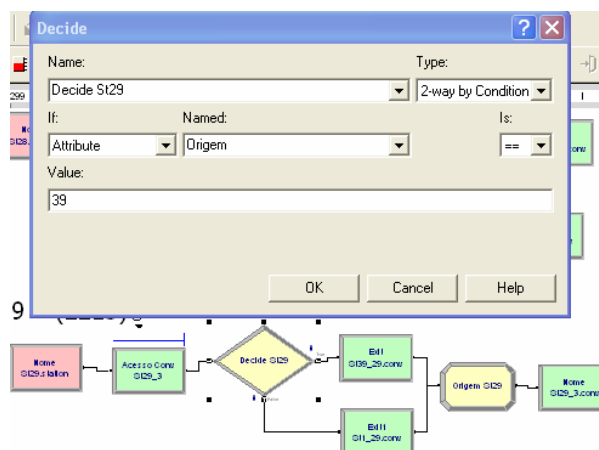


Figura 4-32: Módulo 'decide' na ligação 2E1S

A caixa só irá para um dos módulos 'exit' seguintes. Aqui a caixa sai do respectivo tapete que anteriormente percorria.

No módulo ‘assign’ seguinte é atribuído à caixa o número de origem correspondente ao número do cruzamento (todos os cruzamentos são numerados), para que no cruzamento seguinte se saiba de onde vinha a caixa.

No último módulo do cruzamento a caixa entra no respectivo tapete, o mesmo cujo acesso foi pedido no módulo ‘access’ anterior.

Ligação do tipo 1E2S: Este tipo de ligação (Figura 4-33) indica que só uma caixa poderá chegar à ligação ou seja, o caminho de chegada é único, e a saída da caixa tem duas alternativas

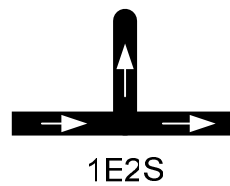


Figura 4-33: Ligação do tipo 1E2S

Existem também inúmeros casos destes no sistema de simulação. Neste tipo de ligação (Figura 4-34), o sistema considera que no cruzamento tem:

- Uma Estação;
- Uma Atribuição com a nova origem da caixa;
- Uma Decisão para verificar qual dos dois tapetes deve aceder;
- Dois grupos de módulos:
 - Um Acesso ao tapete de saída do cruzamento;
 - Uma Saída do tapete de entrada no cruzamento;
 - Um Tapete de saída do cruzamento.

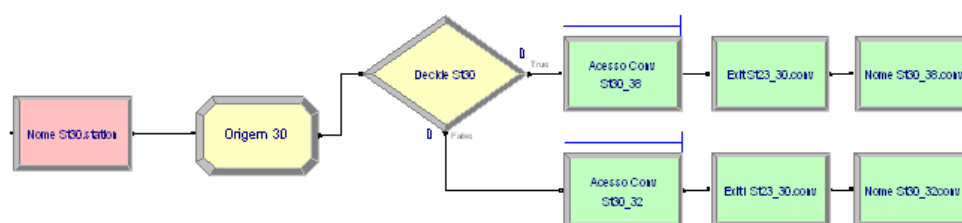


Figura 4-34: Simulação c/ ligação tipo 1E2S hipótese 1

A explicação para este cruzamento é bastante idêntica à da ligação 2E1S com excepção que neste caso o módulo ‘decide’ (Figura 4-35) decide qual o rumo (escolhe um dos tapetes de saída) que a caixa deve tomar. No exemplo da figura, se o destino da caixa for a torre 4 escolhe o tapete que levará à torre 4, senão escolhe o outro tapete.

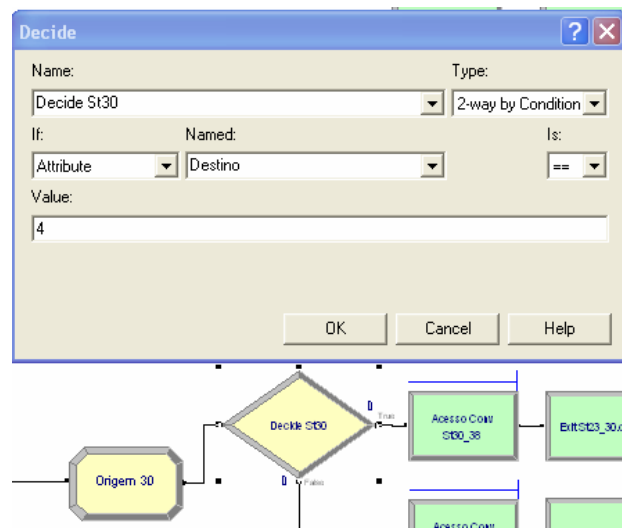


Figura 4-35: Módulo 'decide' na ligação 1E2S – hipótese 1

Nos cruzamentos em que cada um dos dois tapetes de saída pode ter mais que um destino (Figura 4-36) o módulo ‘decide’ tem que necessariamente fazer mais que uma pergunta

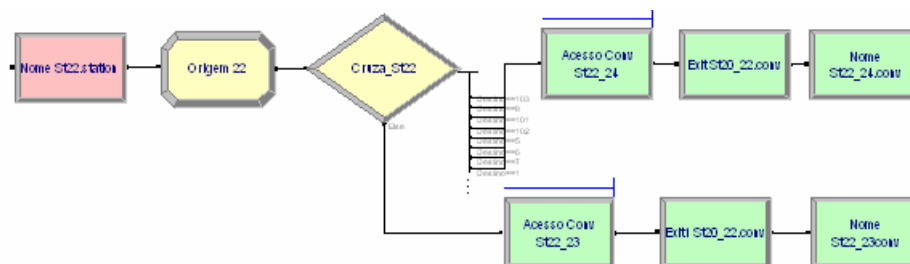


Figura 4-36: Simulação c/ ligação tipo 1E2S - hipótese 2

No módulo ‘decide’ da figura, o sistema faz tantas perguntas quantos os destinos possíveis num dos tapetes de saída. Se os destinos não forem nenhum dos questionados, significa que a caixa irá para um dos outros destinos não mencionados através do outro tapete.

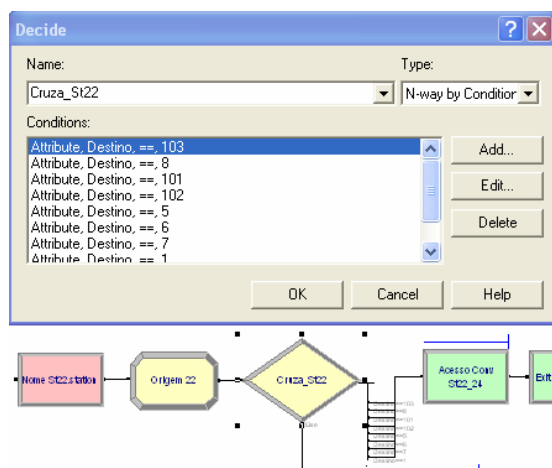


Figura 4-37: Módulo 'decide' na ligação 1E2S – hipótese 2

Ligação do tipo nE1S: Este tipo de ligação (Figura 4-38) é bastante idêntico ao tipo de ligação 2E1S referido anteriormente (Figura 4-30) com a excepção de que depois do módulo 'decide' tem n tapetes possíveis de sair, ver Figura 4-47.

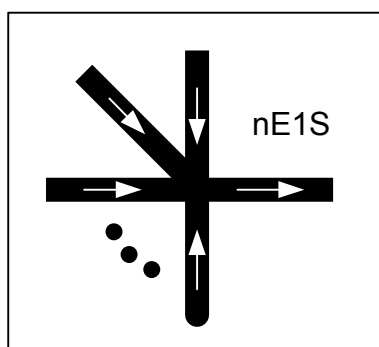


Figura 4-38: Lgação nE1S

Ligação do tipo nE2S: Este tipo de ligação (Figura 4-39) é bastante idêntico ao tipo de ligação nE1S referido anteriormente (Figura 4-38) com a excepção de que depois do módulo 'decide' existe outro módulo 'decide', ver Figura 4-53.

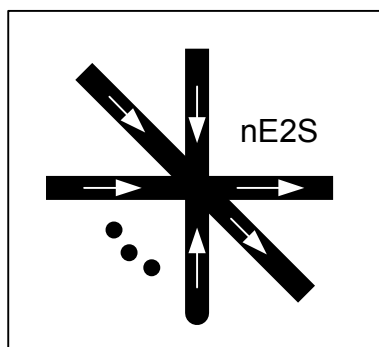


Figura 4-39: Ligação tipo nE2S

Ligação do tipo Entroncamento: Este tipo de ligação no sistema instalado da Codizo, significa que na zona A marcada na Figura 4-40 só anda uma caixa de cada vez. Esta situação não estava prevista no início do projecto instalado na empresa e foi posteriormente colocado mais uma torre (torre 6) o que originou o entroncamento. No sistema de simulação foi necessário decompor este entroncamento em três cruzamentos do tipo 1E2S e 2E1S já analisado anteriormente, colocando 3 estações (nº10, nº18 e nº20). O entroncamento foi então substituído pela seguinte ordem: uma de 2E1S, outra de 1E2S e a última 2E1S.

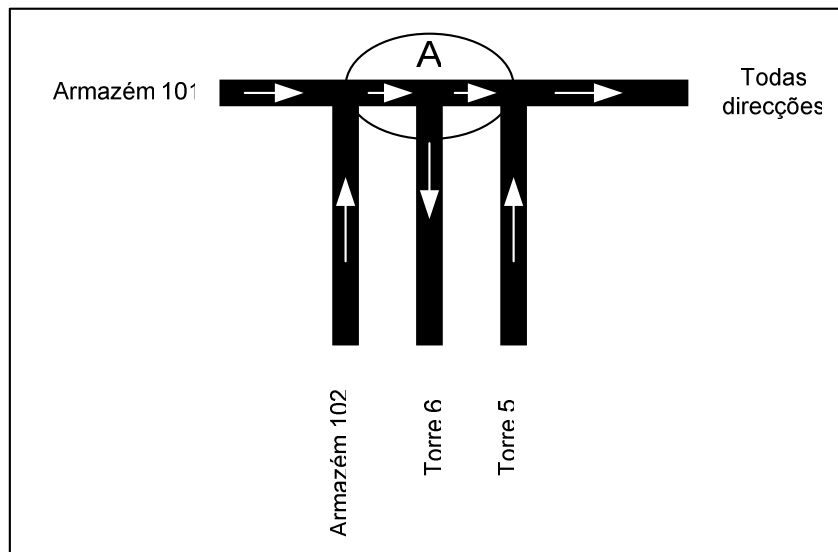


Figura 4-40: Ligação tipo entroncamento

4.2.7.2 Ligações entre tapetes - Armazéns

No capítulo anterior foi indicado que os armazéns por gravidade 101,102 e 103 são compostos por um conjunto de módulos com capacidade para 4 caixas cada, dispostos paralelamente (Figura 3-11). A capacidade do armazém depende do número de módulos.

No sistema de simulação, os armazéns 101, 102 e 103 cada módulo é composto por um tapete que tem início na entrada do armazém (ponto A da Figura 4-41) e termina no local da caixa que fica à frente desse mesmo módulo (ponto C).

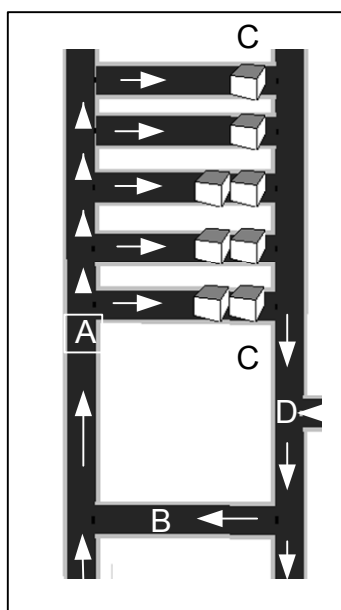


Figura 4-41: Entrada/Saída Armazém

Para as caixas entrarem nos módulos com capacidade de 4 caixas, no ponto A da Figura 4-41 foi colocado uma estação e em cada ponto C (um por cada sub-armazém) foi colocada também uma estação. Todos os tapetes para a entrada dos módulos saem de A e terminam em C.

Para as caixas saírem dos sub-armazéns na direcção das torres ou para voltarem outra vez ao mesmo armazém (ponto B), saem dos tapetes iniciados nas estações colocadas nos pontos C e terminam na estação colocada no ponto D.

Lógica de entrada das caixas nos armazéns por gravidade

O primeiro módulo desta lógica de entradas de caixas nos armazéns por gravidade (Figura 4-42 e Figura 4-43) corresponde ao módulo da estação, ver Figura 4-41 o ponto A.

De seguida a caixa passa pelo bloco VBA, neste bloco é verificado se a caixa deve seguir em frente para outros armazéns, ou deve entrar neste armazém, num dos sub-armazéns de 4 caixas, para isso, verifica qual o sub-armazém para onde se deve dirigir a caixa, corre ainda as funções de pedidos de caixas aos armazéns, função de reentrar, função de recomeçar e verifica se algum dos armazéns atingiu o máximo permitido de caixas.

O módulo seguinte ‘assign’ é utilizado para escrever no atributo a origem da estação (número da estação).

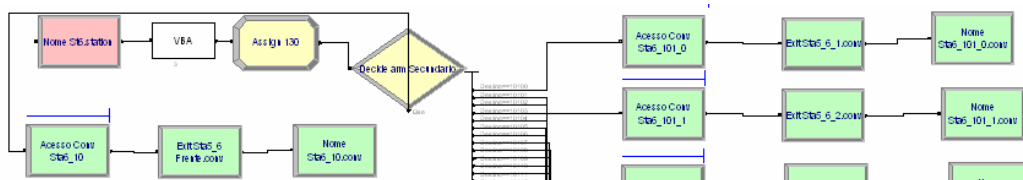


Figura 4-42: Lógica entrada armazém gravidade



Figura 4-43: Armazém gravidade segura caixa

O módulo seguinte ‘decide’ (Figura 4-44) toma a decisão de qual o tapete que a caixa deve tomar para chegar ao sub-armazém.

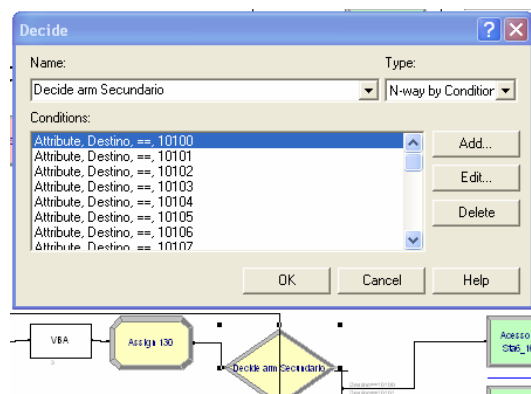


Figura 4-44: Módulo 'decide' na entrada do armazém de gravidade

Os módulos seguintes dependem do que foi decidido no módulo ‘decide’. Estes módulos estão replicados tantas vezes quantos os módulos de 4 caixas que existem no armazém respectivo.

O módulo seguinte ‘access’ pedido para aceder ao tapete que irá dar acesso ao sub-armazém de 4 caixas, decidido no módulo anterior ‘decide’.

O módulo seguinte ‘exit’ corresponde à saída do tapete anterior, o mesmo que trouxe a caixa até à estação referida nesta lógica.

O módulo seguinte ‘convey’ refere-se ao tapete escolhido no módulo ‘decide’ para levar a caixa até ao sub-armazém respectivo.

O módulo seguinte ‘station’ (Figura 4-43) é a estação colocada no respectivo sub-armazém

O módulo seguinte ‘exit’ corresponde à saída da caixa do tapete que a trouxe até à estação.

O módulo seguinte ‘Hold’ é o último módulo da lógica de entrada da caixa no sub-armazém de gravidade. Este módulo segura a caixa até chegar uma ordem para ela sair. Na animação da simulação (Figura 4-41), este módulo fixa a caixa, todas as caixas que chegam a este módulo entram na fila e saem pela ordem de chegada (FIFO), no instante pretendido.

Lógica de saída das caixas nos armazéns por gravidade

A lógica de saída dos armazéns por gravidade está representada nas Figura 4-45 e Figura 4-47.

O primeiro módulo desta lógica de entradas de caixas nos armazéns por gravidade (Figura 4-45) corresponde ao módulo ‘Hold’ com o nome ‘Segura101_0’ que foi precisamente o último módulo da lógica de entrada nos armazéns por gravidade.

As caixas na fila esperam um sinal enviado pela lógica de controlo do sistema (VBA).

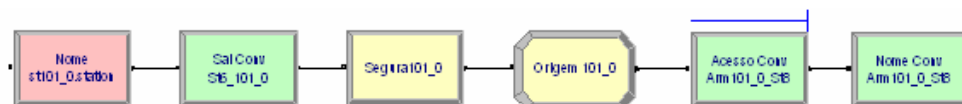


Figura 4-45: Lógica de saída dos armazéns de gravidade

Dependendo do sinal, podem sair da fila do módulo (Figura 4-46) uma ou mais caixas. Como já anteriormente referido o sistema pode pedir uma caixa que está no fim da fila (máximo 4 caixas) e neste caso o sistema de controlo envia um sinal para saírem as 4 caixas todas seguidas.

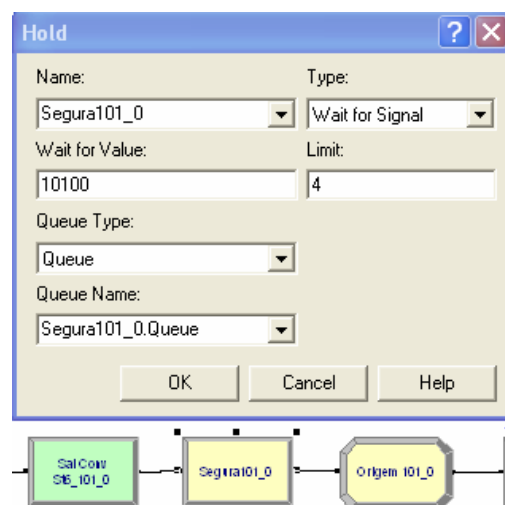


Figura 4-46: Módulo 'Hold' na lógica de saída dos armazéns por gravidade

O módulo seguinte ‘assign’ é utilizado para escrever no atributo origem o armazém e o sub-armazém de onde a caixa está a sair. Deste modo quando a caixa chegar à saída do armazém, este saberá a sua origem (podem ser tantas origens quantos os sub-armazéns que contém o armazém).

No módulo seguinte ‘access’ a caixa realiza o pedido para entrar no tapete que a levará à saída do armazém.

O módulo seguinte ‘convey’ (Figura 4-45) representa o transporte da caixa desde o sub-armazém onde estava até à saída do armazém (Figura 4-41) ponto D.

A lógica na saída do armazém (Figura 4-47) ponto D, tem como primeiro módulo ‘station’ referente à estação de saída do armazém de gravidade 101. Os outros dois armazéns de gravidade 102 e 103 são idênticos, com a excepção da capacidade de caixas (número de sub-armazéns diferentes) e da ligação dos armazéns com os restantes tapetes da logística.

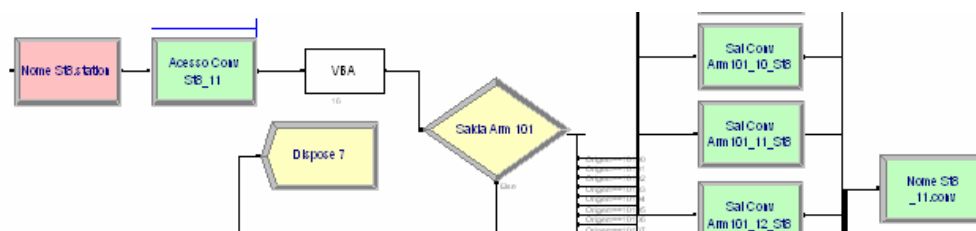


Figura 4-47: Fim da lógica saída dos armazéns de gravidade

O módulo seguinte ‘access’ a caixa realiza o pedido para aceder ao único tapete possível de aceder (ligação do tipo nE1S (Figura 4-38)). O exemplo da Figura 4-47 referente ao armazém, a estação está no cruzamento imediatamente seguinte à saída do armazém enquanto que nos outros dois armazéns de gravidade, a estação não está em nenhum cruzamento.

O bloco seguinte ‘ VBA’ verifica se a caixa deve voltar ao armazém ou deve seguir para uma torre (torre onde está localizada a secção que efectuou o pedido da caixa), neste último caso, se a caixa veio de um dos sub-armazéns e segue em frente, implica que foi uma caixa pedida pelas secções, por isso retira-a da lista dos pedidos. Corre ainda as funções de pedidos de caixas aos armazéns, função de reentrar, função de recomeçar.

O bloco seguinte ‘decide’ verifica em qual dos tapetes (tantos quantos o número de sub-armazéns do armazém de gravidade) trouxe a caixa e encaminha-a para o módulo seguinte ‘exit’.

O bloco seguinte ‘exit’ por onde passa a caixa foi decidido no bloco anterior ‘decide’. Neste bloco a caixa sai do respectivo tapete.

O bloco seguinte ‘convey’ a caixa entra no tapete que a retira da estação na saída do armazém, em direcção a uma torre, ou para voltar ao mesmo armazém.

Lógica de entrada das caixas no armazém num plano horizontal

Na instalação da Codizo o armazém 104 (Figura 3-13 e Figura 3-14) tem 2 tapetes no centro do mesmo ao longo de todo o armazém a deslocarem-se em sentidos contrários.

Na simulação para o armazém 104 foram considerados 3 tapetes no seu interior. Um tapete transporta as caixas até ao centro do armazém (na instalação só é possível as caixas entrarem por um único local) e os outros dois retiram as caixas do armazém, um para cada lado (na instalação é possível as caixas saírem por dois lados diferentes).

Parte da lógica de entrada no armazém está na Figura 4-48, a outra parte está na Figura 4-50. A primeira refere-se à tomada de decisão se vai para um alvéolo do armazém e qual deles, a segunda refere-se ao alvéolo do armazém.

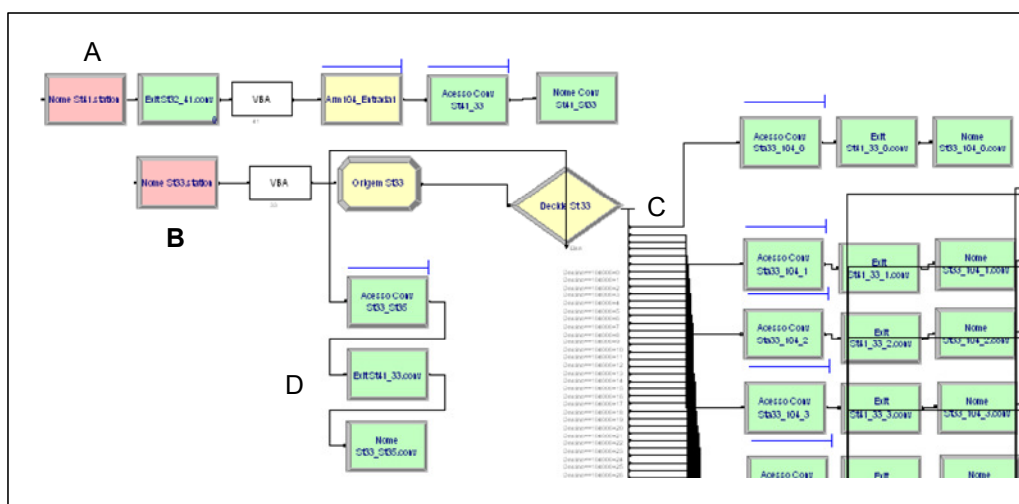


Figura 4-48: Lógica na entrada armazém horizontal

Foi considerado no centro do armazém que haveria 2 estações. Os dois primeiros módulos 'station' da Figura 4-48 ponto A e 'exit' representam a chegada à estação e a saída do tapete utilizado para transportar a caixa

O bloco seguinte 'VBA' contém a lógica que verifica se no bloco seguinte 'hold' estiver lá uma caixa esta deverá avançar. Corre também as funções que verificam se as caixas podem reentrar no sistema ou recomeçar e verifica se os armazéns estão cheios.

O módulo seguinte 'Hold' segura a caixa até receber um sinal para a libertar. No armazém 104 do sistema instalado na empresa a deslocação do centro do armazém para os respectivos alvéolos é realizado por um braço robô de 3 eixos (x,y,z) e só leva e trás para o centro uma caixa de cada vez. Com este módulo 'hold' asseguramos que no sistema de simulação só há movimentação de uma caixa de cada vez, do centro para um alvéolo e vice-versa, tal como no sistema instalado.

Os dois módulos seguintes 'access' e 'convey' permitem o transporte para a estação seguinte da Figura 4-48 ponto B.

O módulo seguinte 'station' ponto B refere-se ao local da tomada de decisão.

O bloco seguinte 'VBA' verifica se o destino da caixa é o do armazém 104. No caso da caixa se dirigir para o armazém 104 e este não estiver cheio, é escrito no atributo da caixa 'destino' o melhor alvéolo para esta. Se a caixa não tiver como destino o armazém 104 ou este estiver cheio, ela é encaminhada para outro destino (destino da caixa ou o armazém mais perto disponível). Actualiza 'arrays' de memória

O melhor alvéolo para a caixa é escolhido sempre como o primeiro livre a partir do centro do armazém para as extremidades, permitindo desta maneira que o braço robô demore o menos tempo possível.

Nos dois módulos seguinte, o primeiro 'assign' escreve a origem da caixa e o segundo 'Decide' (Figura 4-49) direcciona a caixa para o tapete que a levará até à estação (alvéolo) respectiva. Este armazém contém 192 alvéolos.

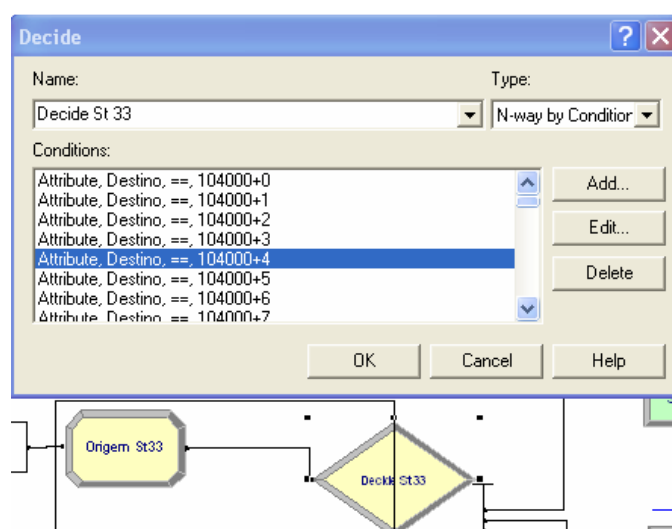


Figura 4-49: Módulo 'Decide' entrada armazém horizontal

Os três módulos seguintes da Figura 4-48 ponto C correspondem ao acesso ao tapete que transportará a caixa para o alvéolo, à saída do tapete anterior e por fim o respectivo tapete.

A lógica referida no ponto D da Figura 4-48 é composta por um bloco 'acesse', um 'exit' e um 'convey'. Conforme referido acima, as caixas que chegam ao armazém e não são para ficar, são direccionadas para a estação de saída de caixas do armazém (station35).

Na Figura 4-50 está a lógica da caixa que chega ao alvéolo, composta pelos módulos 'station', 'exit' 'assign' e 'hold'. No módulo 'exit' a caixa sai do tapete que a trouxe até à estação (alvéolo), no módulo 'assign' é indicado ao sistema que o braço robô

está livre para ser outra vez utilizado (se o braço estiver ocupado não há movimento de outra caixa na direcção a um alvéolo ou de um alvéolo para o centro do armazém).



Figura 4-50: Lógica final entrada armazém horizontal

Por fim o módulo 'hold' (Figura 4-51) que segura a caixa até receber um sinal para largar a caixa.

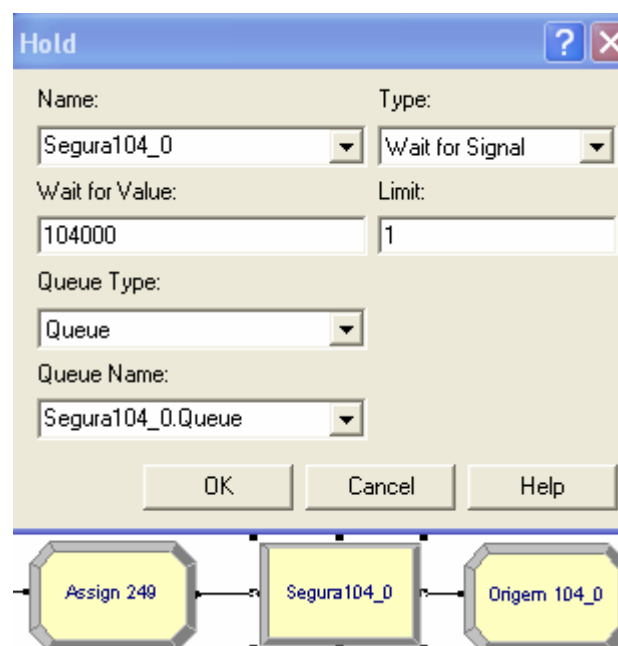


Figura 4-51: Módulo 'Hold' no armazém horizontal

Lógica de saída das caixas no armazém num plano horizontal

A lógica da saída das caixas no armazém horizontal (armazém 104) está representada nas Figura 4-52 e Figura 4-53.

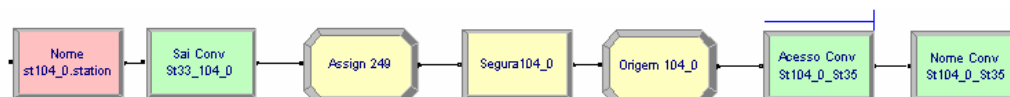


Figura 4-52: Lógica de saída caixas dos alvéolos no armazém horizontal

O primeiro módulo referente à lógica da saída de uma caixa começa no módulo 'hold' (último módulo da lógica de entrada). Este módulo (Figura 4-51) quando recebe um sinal larga a única caixa que tem na sua fila.

O módulo seguinte ‘assign’ indica ao sistema que o braço robô está ocupado e escreve a origem da caixa.

Os dois módulos seguintes ‘access’ pedido de acesso ao tapete e ‘convey’, tapete que transportará a caixa do alvéolo até à estação 35 no centro do armazém.

Os módulos seguintes referem-se a uma ligação do tipo nE2S (Figura 4-39). Na saída das caixas, todos os alvéolos ligam à estação 35, e desta existem duas alternativas para as caixas seguirem: A alternativa das caixas seguirem na direcção das torres 3 e 4, e a alternativa, das caixas seguirem em todas as outras direcções.

O módulo seguinte (Figura 4-53) ‘station’ refere-se à estação 35 utilizada no centro do armazém. Desta estação as caixas saem do armazém.

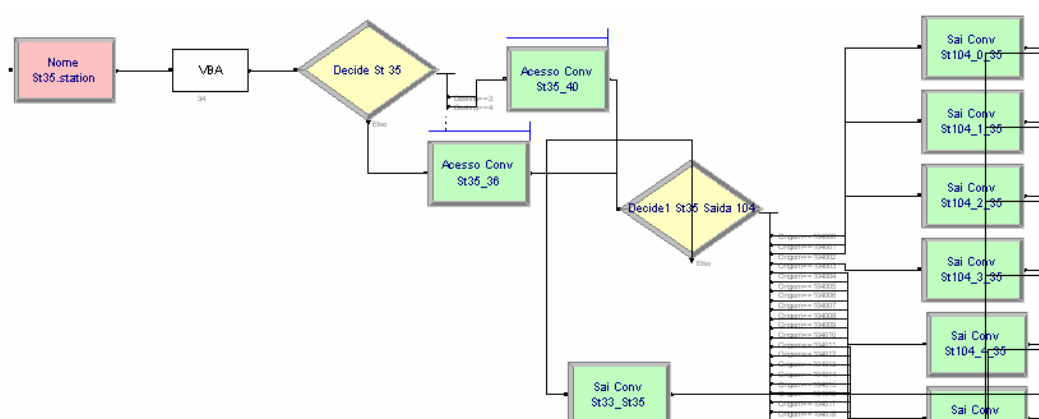


Figura 4-53: Lógica da saída de caixas no armazém horizontal

O bloco seguinte ‘VBA’ verifica se na entrada do armazém, bloco ‘hold’ existe alguma caixa, se houver retira-a do bloco. Coloca o braço do robô desocupado. Retira a caixa da lista de pedidos. Actualiza ‘arrays’ de memória. Corre também as funções que verificam se existem pedidos, se as caixas podem reentrar no sistema ou recomeçar, verifica se os armazéns estão cheios e verifica os buffers de saída das torres (entrada nas secções).

O módulo seguinte, primeiro ‘Decide’ é utilizado essencialmente para verificar qual dos dois tapetes de saída a caixa vai tomar. Verifica se a caixa vai para a torre 3 ou 4 e se assim for, faz o pedido para aceder ao tapete de saída do armazém que levará a caixa nessa direcção. Se a caixa não for para a torre 3 ou 4 então é realizado o pedido para o tapete que levará a caixa no sentido contrário

O módulo seguinte, segundo ‘Decide’ é utilizado essencialmente para decidir por qual dos tapetes a caixa deve sair. Permite verificar a origem do tapete que chegou à estação 35 e conforme o caso sai do respectivo tapete no módulo seguinte ‘exit’

O módulo seguinte ‘Convey’ é o tapete utilizado para retirar a caixa do armazém.

4.2.8 Animação

A animação no Arena é realizada utilizando o módulo ‘convey’ já explicado, e utilizando mais dois módulos de dados, o módulo ‘Conveyor’ e ‘Segment’.

4.2.8.1 O módulo ‘Conveyor’

O bloco ‘Conveyor’ permite a definição de um tapete acumulativo ou não acumulativo para uma caixa em movimento entre estações (‘Stations’).

A velocidade, tamanho da célula, e comprimento dos segmentos, devem ter uma unidade de comprimento que seja comum.

Tapetes não acumulativos, ao sofrerem uma paragem para carregar ou descarregar uma caixa, faz com que todas as caixas parem de forma que o espaçamento entre as caixas no tapete seja constante.

Ao usar um tapete acumulativo, este não sofre paragem ao carregar ou descarregar uma caixa. Se uma paragem (bloqueio utilizando travão pneumático) local ocorrer as caixas continuam a ser conduzidas pelo tapete até à paragem (bloqueio). Neste momento o tamanho da acumulação é avaliado. A soma das distâncias entre as estações ligadas pelo tapete deve ser divisível pelo tamanho da célula. Caso contrário acontecerá um erro de verificação do modelo.

No sistema de simulação desenvolvido, todos os tapetes utilizados são acumulativos, sempre que num local da instalação for accionado um travão no tapete, as caixas param, encostando umas às outras.

Name	Segment Name	Type	Velocity	Units	Cell Size	Max Cells Occupied	Accumulation Length	Initial Status	Report Statistics
St6_101_0.conv	St6_101_0.conv.Seg	Accumulating	1	Per Second	4	1	1	Active	<input checked="" type="checkbox"/>
St6_101_1.conv	St6_101_1.conv.Seg	Accumulating	1	Per Second	4	1	1	Active	<input checked="" type="checkbox"/>
St6_101_2.conv	St6_101_2.conv.Seg	Accumulating	1	Per Second	5	1	1	Active	<input checked="" type="checkbox"/>
St6_10.conv	St6_10.Seg	Accumulating	1	Per Second	10	1	1	Active	<input checked="" type="checkbox"/>

Figura 4-54: Módulo de dados 'Conveyor'

O módulo de dados ‘Conveyor’ (Figura 4-54) tem um conjunto de parâmetros necessários:

Name – Nome do tapete;

Segment name – Nome do segmento utilizado na animação;

Type – Tipo de tapete (acumulativo ou não acumulativo);

Velocity – Velocidade de deslocamento do transportador;

Unit – Unidade de velocidade;

Cell Size – Tamanho da célula;

Max Cells Occupied – Número máximo de células ocupadas;

Accumulation Length – Comprimento da acumulação;

Initial Status – Estado inicial;

Report Statistics – Reporta estatisticamente para efeitos de relatório final.

Na simulação cada caixa ocupa no máximo uma célula. Conjugando o número máximo de células com o tamanho de cada célula com o comprimento do segmento, consegue-se indicar ao sistema de simulação, o comprimento do tapete instalado e o número máximo de caixas permitido em simultâneo no mesmo tapete. Um tapete na simulação não corresponde a um tapete na instalação. Um tapete na simulação liga duas estações entre si.

Por exemplo na instalação a distância entre o local da estação 6 e a estação 10 é de 30 metros num tapete acumulativo com uma velocidade 1m/s. Na simulação (Figura 4-54) na última linha o valor de ‘cell size’ é 10, e o comprimento do tapete na Figura 4-55 é 30, significa para o sistema que nesse tapete só cabem 3 caixas.

4.2.8.2 O módulo ‘Segment’

O módulo “Segment” é utilizado para definir um segmento do percurso de uma caixa num tapete. O percurso, ou local por onde passa o tapete, é composto por uma série de segmentos ligados, onde cada segmento é um ‘link’ directo entre duas estações (‘Stations’) e o caminho é definido por uma estação de início e um conjunto de distâncias até as próximas estações. O Arena na definição do comprimento dos segmentos obriga o utilizador a definir em unidades inteiras. A unidade de medida não é importante, contudo as unidades devem ser consistentes entre os módulos ‘Segment’ e ‘Conveyor’. Os segmentos são unidireccionais de forma que as caixas só se movem entre a estação de início e a próxima estação.

Name	Beginning Station	Next Stations			
Sta4_5_seg	St4.station	1 rows			
Sta3_7.Seg	St3.station	1 rows			
Sta5_6_seg	St5.station	1 rows			
St6_101_0.conv_seg	St6.station	1 rows			
St6_101_1.conv.Seg	St6.station	1 rows			
St6_101_2.conv.Seg	St6.station	1 rows			
St6_10_seg	St6.station	1 rows			
			Next Stations		
				Next Station	Length
			1	st10.station	30

Figura 4-55: Módulo 'Segment'

O módulo de dados ‘Segment’ (Figura 4-55) tem um conjunto de parâmetros necessários:

Name – Nome do segmento;

Beginning Station – Estação de início;

Next Station – Estação seguinte;

Length – Comprimento do segmento

O comprimento do segmento no sistema de simulação é o número de metros de cada segmento. No exemplo mostrado na Figura 4-55 referente ainda ao segmento utilizado entre a estação 6 e a estação 10, na última linha da figura é indicado o comprimento entre a estação 6 e a estação 10.

4.2.8.3 ‘Layout’

Na animação utilizou-se caixas, tapetes e armazéns. Sobre os tapetes e as torres, foram desenhadas as estações e os segmentos conforme as figuras seguintes.

‘Layout’ das estações: Conforme já referido, as estações estão nas ligações perpendiculares dos tapetes, nas torre, nos sub-armazéns e alvéolos, ver Figura 4-56.

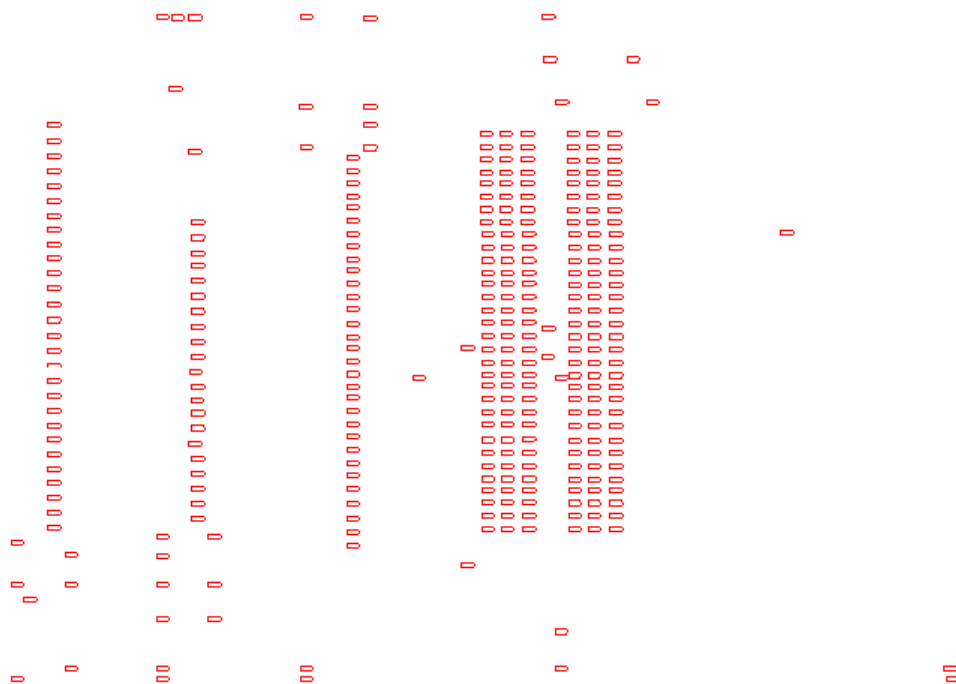


Figura 4-56: ‘Layout’ das estações

‘Layout’ dos segmentos: A ligar todas as estações estão os segmentos (Figura 4-57). No caso do armazém no plano horizontal existe um maior número de segmentos e estações devido à quantidade de alvéolos (para cada alvéolo existe uma estação e dois segmentos)

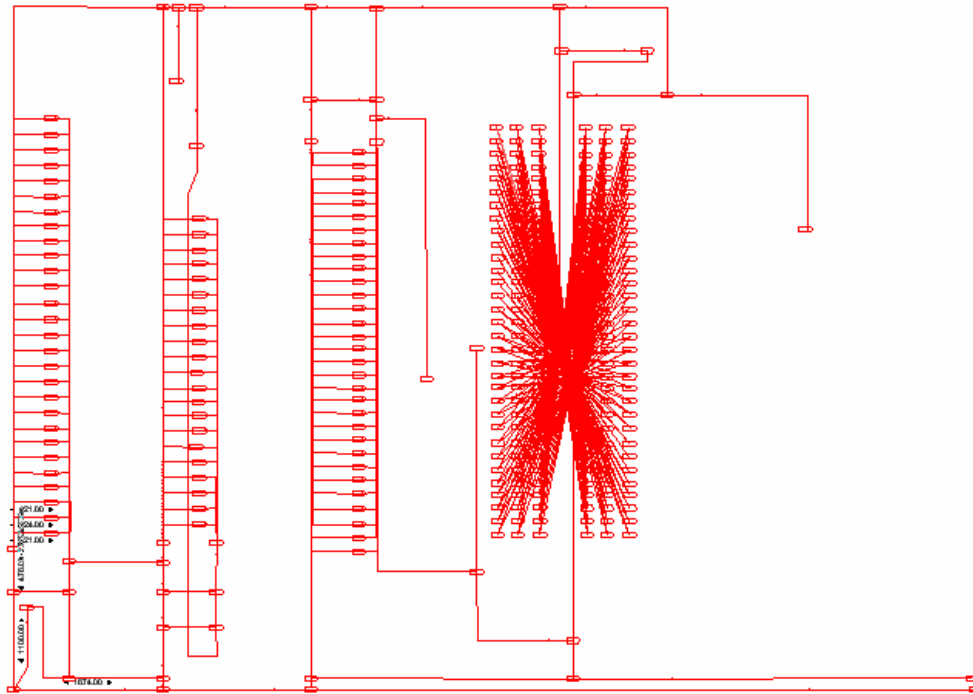


Figura 4-57: ‘Layout’ dos segmentos

‘Layout’ Final: Como trabalho final da simulação, para uma melhor apresentação ao utilizador, em cada segmento está o desenho de um tapete. O utilizador durante o decorrer da simulação, não vê as estações nem os segmentos, ver Figura 4-58, porque o Arena permite que o utilizador utilize um conjunto de “layers” que durante o decorrer da simulação é possível configurar quais os “layers” visíveis e quais os que não são.

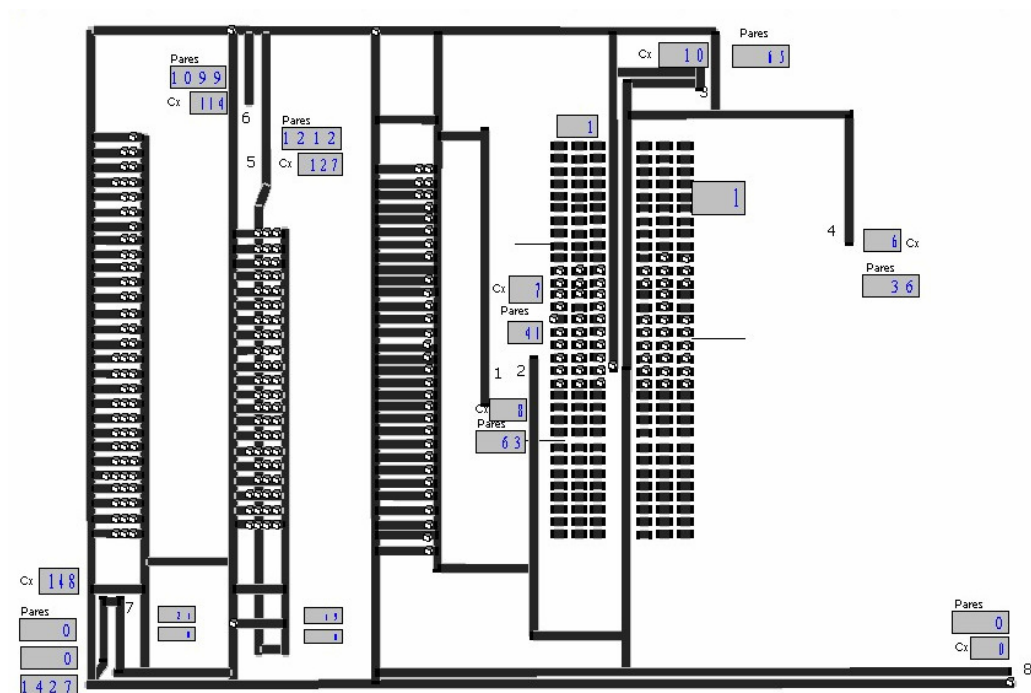


Figura 4-58: 'Layout' Final

5 Resultados

5.1 Introdução

No capítulo anterior apresentou-se o processo de concepção e desenvolvimento do modelo do sistema de movimentação instalado na empresa Codizo utilizando a aplicação de simulação Arena.

Neste capítulo descrevem-se a validação e experimentação do modelo desenvolvido e apresentam-se os resultados obtidos.

5.2 Avaliação dinâmica do sistema em análise

Todo o processo de desenvolvimento de um estudo de simulação, deve passar por um conjunto de fases bem definidas, as quais são referidas e aceites por diversos autores (Law e Kelton (1991), Pedgen, et al. (1995), Banks(1998) são alguns exemplos). Na Figura 5-1 apresenta-se a sequência típica de execução destas fases, conforme sugerida por Law e McComas(1991). Nem todos os estudos seguem rigidamente a execução sequencial destas fases, sendo necessário, por vezes, recolher mais ou menos dados, redefinir o modelo ou mesmo, reformular os objectivos de estudo.

Até à fase 4 da Figura 5-1 tal como apresentado nos capítulos anteriores, sempre que se acrescentava alguma programação ao modelo de simulação, era de seguida, verificado a evolução desse acréscimo, executando o modelo e verificando como este reagia. Contudo só na parte final é que se realizaram os primeiros testes do modelo como um todo, de maneira a validar o mesmo.

5.3 Verificação e validação do modelo

A verificação do modelo de simulação consiste em verificar se o modelo desenvolvido foi implementado correctamente e se o seu funcionamento está conforme o esperado, ou pretendido, por outras palavras é um conjunto de acções cuja meta é certificar se o modelo conceptual foi transcrito de forma adequada quando do uso da linguagem de simulação ou de programação. Consiste em verificar se o programa está a trabalhar como se espera. A verificação tem como objectivo a detecção e remoção de erros, relacionados com a lógica do modelo (Shannon (1998)).

A validação consiste na determinação se o modelo é uma representação aceitável do sistema. O objectivo da validação é passar confiança ao utilizador, mostrando que qualquer execução do modelo irá apresentar resultados que se coadunam com a realidade do sistema estudado. A validação do modelo desenvolvido é conseguida executando o modelo e comparando os seus resultados com os provenientes do sistema real. Se os resultados da execução da simulação se aproximarem dos valores reais, dentro de um nível de confiança desejado, o modelo será validado. A validação do modelo é muito importante, se as simulações são consistentes com a realidade, levam a que tanto o analista como o utilizador, ganhem confiança no modelo.

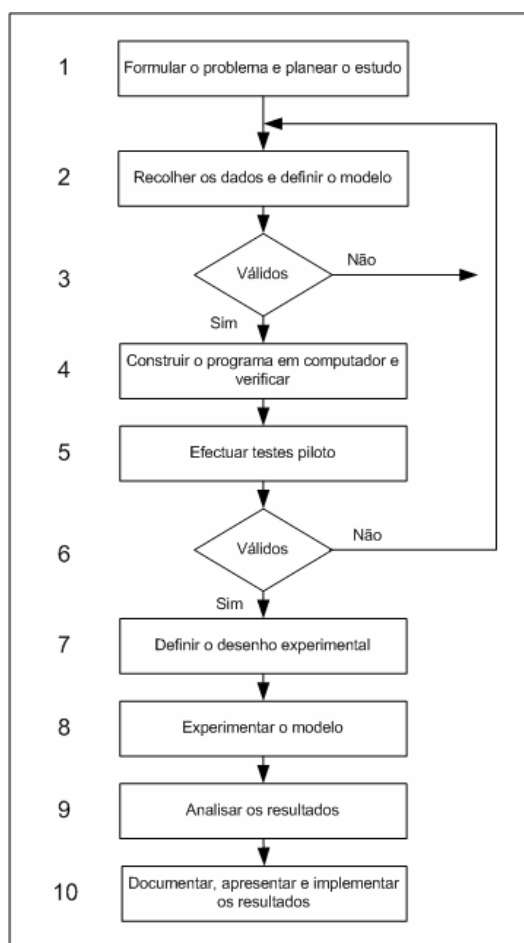


Figura 5-1: Fases de um estudo de simulação (adaptado de Law e McComas (1991))

Durante o processo de validação do modelo, é possível verificar se as abstracções e simplificações introduzidas no modelo conceptual, introduzem erros grosseiros nos resultados. A validade de um modelo está directamente relacionada com os seus objectivos e, de um modo geral, são efectuados diversos testes e avaliações até se obter um grau de confiança suficiente para considerar um modelo válido relativamente aos seus propósitos (Sargent(2000)).

Diversos autores como Law e Kelton (1991), Pedgen, et al. (1995) e Banks (1998), apresentam e descrevem várias técnicas utilizadas na verificação e validação de modelos de simulação. As diferentes técnicas deverão ser usadas conjuntamente uma

vez que, a sua utilização individual, não garante a correcta implementação do modelo e a sua adequação aos objectivos propostos (Ramos (2001)).

As principais técnicas e acções de verificação utilizadas neste estudo foram as seguintes:

- Controlar a execução interactivamente ('debugger'): o controlador de execução possibilitou a monitorização do estado das variáveis e o controlo da execução do modelo de um modo interactivo, ou seja, permitiu executar o modelo passo a passo, suspender a execução do modelo em pontos críticos, e desta maneira permitem analisar os dados e resultados, alterar o valor de determinadas variáveis para forçar a ocorrência de erros. Para ajudar no "debugger" e permitir de uma forma mais rápida a alteração de algumas variáveis mais críticas foi adicionado um écran (Figura 5-2) que aparece sempre no arranque do programa e sempre que o utilizador interrompe o programa para verificar o estado das variáveis e resultados.

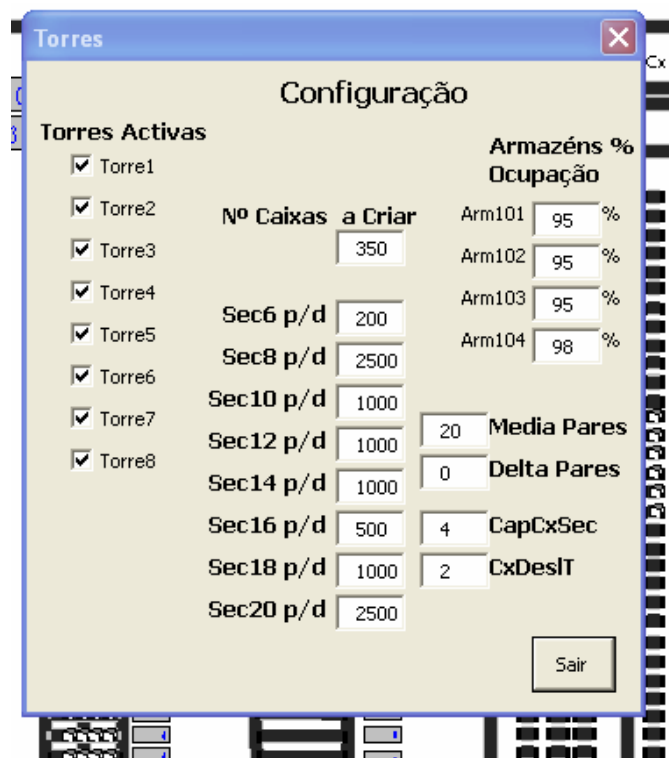


Figura 5-2: Ecrã de arranque e paragem do modelo

O utilizador pode desta maneira interagir como a seguir descrevo:

- 1- Activar/Desactivando as torres permitindo que estas peçam ou não peças, caixas aos armazéns;
- 2- Alterar o número de caixas criadas pela simulação;
- 3- Alterar a percentagem de ocupação permitida para cada armazém;

- 4- Alterar o número de pares médio em cada caixa;
 - 5- Alterar o número de caixas a deslocarem-se para as torres em simultâneo.
- Teste de valores de entrada: a alteração dos valores dos parâmetros de entrada na execução do modelo permitiu avaliar o comportamento do modelo e verificar como se comportavam os parâmetros de saída, sobretudo com valores de entrada extremos ou errados. O écran apresentado na Figura 4-52 contribuiu em boa medida para esses testes.
 - Observar a animação: através da animação, houve a possibilidade de detecção de determinados erros relacionados com a interacção dos diferentes componentes do sistema através da visualização dinâmica do comportamento do modelo desenvolvido durante a execução do mesmo.

Se o modelo de simulação é complexo, ou se o sistema a modelar não existe, ou não está instalado, a dificuldade em validar o modelo de simulação aumenta. Neste estudo as dificuldades foram um pouco amenizadas apesar do sistema modelado ser relativamente complexo, pelo facto de haver um sistema real já instalado. Contudo, a existência da instalação ajudou alguma coisa mas não muito, porque durante a fase do presente estudo, o sistema instalado movimentou muito menos caixas do que as inicialmente previstas.

As principais técnicas ou acções empregues segundo Ramos (2001), e adaptadas para o presente caso de estudo, para validar o modelo desenvolvido consistiram em :

- Análise de sensibilidade: consistiu na alteração dos valores dos parâmetros de entrada e analisar a sensibilidade (comportamento) dos parâmetros de saída.
- Testes piloto e teste de ‘turing’: a utilização de testes piloto, fase 5 da Figura 5-1, permitiu um tipo de validação subjectiva teste “Turing”, que consiste, na comparação dos dados obtidos pela simulação com os dados obtidos pelo sistema real, num mesmo formato. Posteriormente a apresentação dos dois resultados apresentados aos especialistas do sistema em forma de tabelas de nº de caixas para cada secção. Uma outra avaliação subjectiva, seria o recurso de especialistas “expert”, consistindo em submeter os dados gerados, a especialistas notórios da área. Estes baseados em suas experiências emitiriam pareceres sobre a confiabilidade do modelo. Este tipo de validação é aplicado quando não existe o sistema real, ou quando este não é utilizável (Silva (2005)), o que não foi o caso de estudo.
- Validade do analista: a execução de várias execuções do modelo, permitindo verificar a variabilidade estocástica do modelo.
- Análise com a equipa da empresa: o modelo conceptual de simulação e o seu comportamento foram discutidos com os membros da organização e foram avaliadas a sua razoabilidade.

- Animação do sistema modelado: a observação, por parte dos especialistas do modelo verificando o desempenho ao longo da execução da simulação permitiu contribuir para a validade do modelo desenvolvido.

No processo de validação do modelo foram verificados vários pontos considerados importantes para os quais a animação contribuiu significativamente:

- Entrada de caixas no sistema;
- A entrada e saída de caixas nos armazéns de gravidade;
- Armazéns de gravidade – ocupação dos subarmazéns;
- A entrada e saída de caixas no Armazém com o robô;
- Armazém com robô – ocupação dos alvéolos;
- Ocupação dos armazéns (taxa de ocupação);
- Cruzamentos – verificação da correcta direcção das caixas;
- ‘Buffers’ de entrada e saída de caixas nas torres;
- Saídas das torres (incrementos dos apontadores das caixas nas gamas operatórias, tempos da operação seguinte e destino da caixa);
- Verificar o algoritmo de pedido de caixas;
- Visualização dos atributos das caixas no espaço temporal da execução do modelo.

A utilização do conjunto de técnicas atrás referidas permitiu confirmar um correcto desenvolvimento do modelo de simulação e possibilitou a avaliação da sua precisão e adequação relativamente aos objectivos inicialmente definidos. A verificação e validação do modelo permitiram também ganhar confiança sobre os valores das variáveis de saída (resultados) sendo possível prosseguir o estudo e passar às fases de experimentação e análise de resultados.

5.4 Experimentação e análise de resultados

A experimentação de um modelo de simulação é a fase na qual as diversas alternativas serão testadas. Além disso, podem ser traçados os cenários para análise de sensibilidade do tipo “what-if”, para avaliar os efeitos de possíveis alterações.

A experimentação consiste na definição de vários cenários e na identificação de um conjunto de parâmetros relacionados com as condições de testes a realizar. Os

parâmetros referidos, estão relacionados com o tipo de simulação considerado (Banks (1998)):

- a) Simulação sem uma condição natural de paragem (Simulação não terminal ou de estado estacionário): O sistema não possui uma condição natural de paragem (evento) e a sua duração não é finita, ou seja, funciona continuamente ou ao menos por um período muito longo. O instante final $t=TF$ não está determinado pela natureza do problema, senão é mais um parâmetro a ser determinado no desenho do experimento. Isto não significa que a simulação nunca termina, mas que, teoricamente, a simulação pode continuar infinitamente sem afectar as saídas (resultados). Utiliza-se quando se quer estudar características que não dependam do estado inicial no instante $t=0$. O objectivo principal deste tipo de simulação é a avaliação do comportamento estacionário do sistema e, como tal, os efeitos das condições iniciais devem ser removidos dos resultados da simulação.

- b) Simulação com uma condição natural de paragem (Simulação terminal ou de estado transitório): O sistema possui uma condição inicial fixa e uma ocorrência que determina a duração da simulação. A simulação frequentemente repete um ciclo. A simulação é executada durante um certo tempo TF , onde F é um evento (ou conjunto de eventos). O sistema simulado começa a funcionar por exemplo no instante $t=0$ e termina no $t=TF$. As condições iniciais são especificadas. O objectivo principal deste tipo de simulação é a avaliação do comportamento do sistema para um período fixo típico e, uma vez que as condições iniciais e de paragens são fixas o único factor controlável é o número de replicações.

A escolha do tipo de simulação mais apropriado para o sistema em causa condiciona a definição das condições experimentais do modelo de simulação:

- Condições iniciais para cada replicação;
- A quantidade de execuções independentes do modelo (replicações);
- Duração da simulação;
- Duração do período de aquecimento (“Warm-up”).

No estudo apresentado, concluiu-se que o sistema analisado corresponde a uma simulação não terminal ou de estado estacionário, as características a estudar não dependem do estado inicial no instante $t = 0$, o objectivo principal deste tipo de simulação é a avaliação do comportamento estacionário do sistema. Apesar de neste tipo de simulação o tempo final TF não estar determinado pela natureza do problema, foi considerado um dia de trabalho (480 minutos).

Na experimentação do modelo de simulação foi considerado:

- Análise de sensibilidade;
- Comparação de cenários;

5.4.1 Análise de sensibilidade

Para realizar uma análise de sensibilidade foram considerados vários pontos a referir:

1. Seleccionar o conjunto de variáveis de entrada e parâmetros do sistema que maior impacto causa às variáveis de saída;
2. Seleccionar as variáveis de saída de maior importância;
3. Executar o modelo para as combinações dos valores das variáveis de entrada e parâmetros do sistema;
4. Quantificar o impacto causado nas variáveis de saída.

5.4.1.1 Seleccionar o conjunto de variáveis de entrada e parâmetros do sistema que maior impacto causam nas variáveis de saída

Na selecção do conjunto de variáveis de entrada e parâmetros do sistema que maior impacto causam nas variáveis de saída, foram consideradas as seguintes variáveis:

- TotalCaixasCriada – Número de caixas que o sistema de simulação cria no sistema. Na lógica do sistema foi considerado que sempre que uma caixa acabava de realizar as operações da gama operatória é colocada automaticamente na torre 8 para entrar novamente no sistema, e neste caso fica na fila de espera “Fila Recomeça”, ver Figura 4-12. Significa então que no sistema o número máximo de caixas em simultâneo é igual a esta variável.
- MediaPares – Número mínimo de pares em cada caixa. Esta variável é carregada no arranque do sistema.
- DeltaPares – Número máximo de pares a acrescentar à variável ‘MediaPares’. O somatório desta variável e da variável ‘MediaPares’ constitui o número de pares que uma caixa transporta no sistema. Desta forma o número de pares de cada caixa são valores aleatórios gerados no VBA utilizando a função RND (distribuição rectangular ou uniforme) e são gerados todos os números antes de entrar a primeira caixa no transportador (arranque do sistema).
- Sec12ParesDia – Número de pares pretendidos produzir na secção 12 (Torre 1). Esta variável influencia bastante, na medida em que está directamente relacionada com o tempo previsto para cada caixa nesta secção.
- Sec14ParesDia; Sec18ParesDia; Sec16ParesDia; Sec8ParesDia; Sec10ParesDia; Sec6ParesDia; Sec20ParesDia – O mesmo significado que a ‘Sec12ParesDia’ mas, para as restantes secções nas outras torres.

- CapCxSec – Capacidade máxima de caixas dentro das secções. No modelo só uma caixa é processada de cada vez, as restantes (até à capacidade máxima), caso o sistema consiga abastecer a secção, ficam na fila do processo das secções.
- CxDeslTorres – Número de caixas máximo possível de se deslocar em simultâneo para cada uma das torres.
- DesvPadrão – Desvio padrão da distribuição Normal utilizada nos processos de cada uma das secções, ver Figura 4-25.
- “Number of Replication” – número de replicações independentes a utilizar
- “Warm-up Período” – Período de tempo de execução da simulação em que não são recolhidas as estatísticas do mesmo. Tempo de aquecimento.
- “Replication Length” - Tempo de duração da execução da simulação do modelo. Este tempo engloba o tempo “Replication Length” referido anteriormente.
- “Hours Per Day” – Número de horas consideradas para um dia de trabalho.

5.4.1.2 Seleccionar as variáveis de saída de maior importância

Na selecção do conjunto de variáveis de saída de maior importância foram consideradas as seguintes variáveis:

- “ResourceT1_Sec12.Utilization” – taxa de utilização dos recursos da secção 12 (Torre1) . O recurso utilizado na secção do modelo de simulação, representa os recursos (totalidade dos operadores) na mesma secção. Com a análise desta variável será possível verificar se o sistema de movimentação e armazenamento consegue abastecer a secção sem que esta pare.
- ‘ResourceT2_Sec14.Utilization’; ‘ResourceT3_Sec18.Utilization’; ‘ResourceT4_Sec16.Utilization’; ‘ResourceT5_Sec8.Utilization’; ‘ResourceT6_Sec10.Utilization’; ‘ResourceT7_Sec6.Utilization’; ‘ResourceT8_Sec20.Utilization’ - O mesmo significado que a variável ‘ResourceT1_Sec12.Utilization’ mas, para as restantes secções nas outras torres.
- ProcessaT1_sec12 – Número de caixas realizadas produzidas na secção 12 (Torre1) durante o período estipulado (“Replication Length” - “Replication Length”) indica a quantidade de caixas totalmente processadas nesta secção.
- ProcessaT2_sec14; ProcessaT3_sec18; ProcessaT4_sec16; ProcessaT5_sec8; ProcessaT6_sec10; ProcessaT7_sec6; ProcessaT8_sec20 - O mesmo

significado que a variável ‘ProcessaT1_sec12 mas, para as restantes secções na outras torres.

5.4.1.3 Executar o modelo para as combinações dos valores das variáveis de entrada e parâmetros do sistema

Na fase de experimentação do modelo de simulação consideraram-se as seguintes condições:

- A duração da simulação corresponde a um dia de trabalho e reflecte o período de planeamento típico para um dia.
- Efectuaram-se dez replicações (n=10), cinco replicações (n=5) e ainda uma replicação (n=1) independentes do modelo. O número de replicações foi definido através da execução de testes piloto e da avaliação da variabilidade associada aos resultados obtidos para diferentes medidas de desempenho.
- As condições iniciais para cada replicação são caracterizadas pelo sistema parado e quase vazio (só 8 caixas no armazém 101).
- Foi considerado um período de aquecimento de 60 minutos, uma vez que as condições iniciais (parado e vazio) não são representativas do seu normal funcionamento e por esta razão considerou-se que para estimar as medidas de desempenho pretendido as secções deveriam estar em pleno funcionamento. Os operadores nas secções param no fim do dia de trabalho, e retomam no dia seguinte no ponto onde pararam, por isso, os dias são repetitivos. O tempo de 60 minutos para aquecimento é o tempo suficiente para chegarem caixas a todas as secções.
- Combinação das variáveis de entrada referentes ao número de pares previstos para realizar em cada secção. Os valores iniciais para cada uma das secções, ver Tabela 5-1.

Torre	T1	T2	T3	T4	T5	T6	T7	T8
Secção	12	14	18	16	08	10	06	20
Pares	1000	1000	1000	500	2500	1000	200	2500

Tabela 5-1: Quantidade de pares pretendidos em cada secção

O número de pares na Tabela 5-1, é o número de pares inicialmente pretendidos para realizar em cada secção na altura em que o sistema de movimentação e armazenamento foi instalado na empresa Codizo.

Parâmetros/Variáveis	Valores
CapCxSec	2
CxDeslTorres	2
MediaPares	10
DeltaPares	0
DesvPadrão	0.2
TotalCaixasCriada	250
Número de replicações	10
Período de aquecimento	60 Minutos
Tempo da replicação	540 Minutos
Horas por dia	24 Horas

Tabela 5-2: Parâmetros da simulação

- Gamas operatórias em simultâneo: quatro gamas operatórias diferentes foram criadas no arranque do sistema e associadas às caixas. Para que o sistema ficasse equilibrado as quatro gamas operatórias diferentes, foram distribuídas por sete gamas operatórias sendo algumas repetidas. Desta forma, ao ser criada no sistema a caixa 1 esta fica, com a gama G2.1(Tabela 5-3), a caixa 2 fica com G2.2, a caixa 3 com a G2.3, a caixa 4 com a gama G2.4, a caixa 5 com a gama G2.5, a caixa 6 com a gama G2.6, a caixa 7 com a gama G2.7, a caixa 8 com a gama G2.1 e sucessivamente. Por cada caixa para a costura, iriam sete caixas para a premontagem, três para montagem 1, três para montagem 2, três para enformar2 e uma para montagem 3.

	Secções – seq 1	Secções – seq 2	Secções – seq 3	Secções – seq 4
G2.1	Costura (sec6)	Premontagem(sec8)	Moldar(sec10)	Montagem1(sec12)
G2.2	Premontagem(sec8)	Moldar(sec10)	Montagem1(sec12)	
G2.3	Premontagem(sec8)	Moldar(sec10)	Montagem1(sec12)	
G2.4	Premontagem(sec8)	Enformar2(sec18)	Montagem2(sec14)	
G2.5	Premontagem(sec8)	Enformar2(sec18)	Montagem2(sec14)	
G2.6	Premontagem(sec8)	Enformar2(sec18)	Montagem2(sec14)	
G2.7	Montagem3(sec16)			

Tabela 5-3: Gamas operatórias no sistema tipo G2

Este equilíbrio de entrada de caixas para cada secção previsto é suficiente, porque as caixas sempre que realizam a última operação voltam a reentrar no sistema e desta maneira é garantido que as secções não são abastecidas por falta de caixas nos armazéns.

O conjunto de gamas apresentado na Tabela 5-3, é um exemplo possível. Noutro conjunto de gamas operatórias, o sistema não se comporta da mesma maneira.

O modelo de simulação foi executado várias vezes para 1, 5 e 10 replicações, alterando os valores dos parâmetros de simulação referentes a capacidade de caixas nas secções, número máximo de caixas possíveis de se deslocarem na direcção das torres, número de pares em cada caixa.

5.4.1.4 Quantificar o impacto causado nas variáveis de saída.

A alteração das variáveis de entrada e dos parâmetros do sistema de simulação causa impacto nas saídas, basta afectar um dos valores de entrada para se verificar alterações significativas nos valores de saída.

As medidas de desempenho estão associadas a um dos seguintes elementos da simulação:

- Entidade (Entity);
- Localização (Location);
- Estação (Station);
- Fila (Queue);
- Recurso (Resource);
- Variável (Variable);
- Atributo (Attribute);
- Processo (Process).

No caso em estudo, foi dada mais importância à Entidade (caixas), ao Recurso e ao Processo.

Como referido anteriormente, o estudo apresentado, corresponde a uma simulação não terminal, ou do estado estacionário, porque continuando a simulação por mais que um dia os resultados não se alteravam, contudo considerou-se que a simulação deveria decorrer durante um dia, 8 horas de trabalho (480 minutos). Considerou-se a análise num conjunto de observações independente, através da realização de várias replicações iniciadas com diferentes sementes de números pseudo-aleatórios (Sanchez(1999)).

Segundo Ayres (2000), para os sistemas não terminais, a execução da simulação é feita dum modo contínuo. Contudo na maioria dos sistemas, estes não se iniciam “vazios”, ou seja, sem caixas ou sem pessoas. Para as estatísticas serem válidas é necessário o sistema estar em funcionamento quando elas começam a ser recolhidas. Para resolver este problema, logo que a simulação inicia, é estabelecido um tempo de “warm-up”, ou aquecimento para o qual o Arena não recolhe informações. O tempo de aquecimento, também varia conforme o modelo, e deve ser determinado por tentativas (Kelton e Sadowski (1998)). Passado o tempo de aquecimento a simulação pode ser processada da mesma maneira que são os sistemas terminais. Contudo o tempo de aquecimento só deve ser aplicado à primeira replicação, o sistema nas seguintes não deve ser reiniciado (Kelton e Sadowski (1998)). No caso em estudo as caixas permanecem no sistema e a replicação seguinte inicia-se já com elas.

Intervalo de confiança: Depois de calculada a média e o desvio padrão, baseado numa quantidade de amostras de n execuções da simulação, poderemos estar interessados em saber quanto próximo da média verdadeira, está a média das execuções. O método utilizado para isso é a estimativa do intervalo de confiança. Um intervalo de confiança, é um intervalo na qual podemos ter um certo nível de confiança de que a média se encontra lá (Costa (2002)).

O Arena gera um conjunto de relatórios onde é possível verificar os valores mínimos, médios, máximos e “half-width” (meio comprimento do intervalo de confiança), das observações verificadas para as diferentes medidas de desempenho de um sistema. O Arena segundo Ramos(2001), também constrói intervalos de confiança a 95% (valor por defeito) para o valor esperado das medidas, ou seja, o risco (erro) é 0,05 (5%). A estimação por intervalo possibilita a obtenção de informação relativa ao rigor ou confiança das estimativas associadas aos diferentes parâmetros do sistema. O nível de confiança do intervalo (95%) corresponde à probabilidade de este incluir o parâmetro estimado (Guimarães e Cabral (1997)).

Segundo Costa (2002), o “half-width” é a medida de precisão, obtida a partir da amostra (número de replicações). Quanto menor for este valor, melhor a estimativa da média. Este valor pode ser reduzido, aumentando a quantidade de observações da amostra, por outras palavras, aumentando o número de replicações ($u = [x-h, x+h]$) sendo u a média, x a estimativa da média e h o semi-intervalo (meio comprimento do intervalo de confiança).

Seguidamente apresentam-se os resultados (em termos de valor médio para as cinco e dez replicações efectuadas) para as medidas de desempenho seleccionadas. Considerando os pares pretendidos para cada secção apresentados na Tabela 5-1, com as variáveis e parâmetros apresentados na Tabela 5-2 e com as gamas operatórias apresentadas na Tabela 5-3.

1 replicação								
Torres	T1	T2	T3	T4	T5	T6	T7	T8
Secções	12	14	18	16	8	10	6	20
Taxa de Utilização	0.992	0.993	1.000	1.000	0.998	1.000	1.000	0.000

Tabela 5-4: Taxa de utilização dos recursos homem nas secções c/ uma replicação

1 replicação								
Torres	T1	T2	T3	T4	T5	T6	T7	T8
Secções	12	14	18	16	8	10	6	20
Quantidade de Cx	100	99	100	50	251	100	20	0

Tabela 5-5: Número de caixas realizadas em cada secção c/ uma replicação

5 replicações								
Torres	T1	T2	T3	T4	T5	T6	T7	T8
Secções	12	14	18	16	8	10	6	20
Taxa de Utilização	0.996	0.990	1.000	1.000	0.997	1.000	1.000	0.000

Tabela 5-6: Taxa média de utilização dos recursos homem nas secções c/ cinco replicações

5 replicações								
Torres	T1	T2	T3	T4	T5	T6	T7	T8
Secções	12	14	18	16	8	10	6	20
Quantidade de Cx	100	99.6	99.8	49.8	249.4	100.2	20.2	0

Tabela 5-7: Média de caixas realizadas em cada secção c/ cinco replicações

10 replicações								
Torres	T1	T2	T3	T4	T5	T6	T7	T8
Secções	12	14	18	16	8	10	6	20
Taxa de Utilização	0.997	0.992	1.000	1.000	0.996	1.000	1.000	0.000

Tabela 5-8: Taxa média de utilização dos recursos homem nas secções c/ dez replicações

10 replicações								
Torres	T1	T2	T3	T4	T5	T6	T7	T8
Secções	12	14	18	16	8	10	6	20
Quantidade de Cx	99.8	99.3	99.8	50	249.7	100.3	20.1	0

Tabela 5-9: Média de caixas realizadas em cada secção c/ dez replicações

Os resultados obtidos através do estudo de simulação permitem à empresa avaliar alguns parâmetros relacionados com o comportamento dinâmico do sistema de movimentação e armazenamento.

É perceptível que os valores apresentados demonstram que o sistema está suficientemente dimensionado para os pares pretendidos a realizar em cada secção na empresa, os operadores nunca param para os pares pretendidos.

5.4.2 Comparação de cenários

A simulação, a partir de situações iniciais diferentes, oferece linhas de acção alternativas para o agente de decisão. Para realizar uma comparação entre cenários foram considerados vários pontos a referir:

5.4.2.1 Escolha dos cenários a comparar

Vários cenários foram colocados, utilizando a ferramenta “Process Analyser” incluída na Arena Profissional. O Arena grava num ficheiro de programa do modelo com extensão (*.p), e com base neste ficheiro, o “Process Analyser” executa o modelo para determinadas variáveis de entrada e parâmetros do sistema (“Controls”) que maior impacto causa às variáveis de saída (“Responses”) para as gamas operatórias G2.x (já referidas em 5.4.1 na Tabela 5-3 nuns cenários e as mesmas variáveis de entrada e saída para as gamas operatórias em simultâneo G1.x indicadas na Tabela 5-10.

	Secções – seq 1	Secções – seq 2	Secções – seq 3	Secções – seq 4
G1.1	Costura (sec6)	Premontagem(sec8)	Moldar(sec10)	Montagem1(sec12)
G1.2	Premontagem(sec8)	Enformar2(sec18)	Montagem2(sec14)	
G1.3	Premontagem(sec8)	Subcontratados(sec20)	Montagem3(sec18)	
G1.4	Moldar(sec10)	Montagem1(sec12)		
G1.5	Enformar2(sec18)	Montagem2(sec14)		
G1.6	Montagem3(sec6)			
G1.7	Premontagem(sec8)	Moldar(sec10)	Montagem1(sec12)	
G1.8	Enformar2(sec18)	Montagem2(sec14)		
G1.9	Moldar(sec10)	Montagem1(sec12)		
G1.10	Enformar2(sec18)	Montagem2(sec14)		

Tabela 5-10: Gamas operatórias no sistema tipo G1

5.4.2.2 Configuração dos modelos para cada um dos cenários

As gamas operatórias G1 e G2 são introduzidas no modelo através de programação (VBA) e os parâmetros: “Number of Replication”; “Warm-up Periodo”; “Replication Length” e “Hours Per Day” são alterados utilizando o set up do Arena, as restantes variáveis de entrada e parâmetros dos modelos podem ser alterados na aplicação “Process Analyser”. A combinação dos 14 valores possíveis de alterar utilizando o “Process Analyser” é bastante elevada. Por este facto, foi realizado um conjunto bastante alargado de combinações, alterando as variáveis que mais impacto causavam na saída, mas não esgotou todas as combinações possíveis.

5.4.2.3 Comparação dos resultados nos vários cenários

O modelo foi executado com diferentes valores para as variáveis de entrada e parâmetros do modelo. Seguem-se algumas tabelas com o resumo de alguns cenários. A análise comparativa dos cenários não pretende ser exaustiva. Não é objectivo deste estudo otimizar o fluxo de caixas em cada secção, mas sim o estudo do sistema de logística interna, para um conjunto de cenários a colocar pelo utilizador (variação das variáveis de entrada).

Modelo G1; Período de aquecimento = 480 minutos, Tempo de análise = 480 minutos (1 dia de trabalho); 10 replicações:

Caixas	Média Pares	Delta Pares	Cap Cx	Cx Des	Des Pad	Sec12 T1	Sec14 T2	Sec18 T3	Sec16 T4	Sec8 T5	Sec10 T6	Sec6 T7	Sec20 T8
250	10	0	2	2	0.20	900	900	900	500	900	900	200	1800
Taxa de utilização						0.998	0.997	1.000	1.000	1.000	1.000	1.000	0.126
Quantidade de caixas						90.00	89.70	90.30	50.10	90.20	89.80	19.90	22.70

Tabela 5-11: Gama tipo 1, com 10 pares por caixa

Caixas	Média Pares	Delta Pares	Cap Cx	Cx Desl	Des Pad	Sec12 T1	Sec14 T2	Sec18 T3	Sec16 T4	Sec8 T5	Sec10 T6	Sec6 T7	Sec20 T8
250	5	10	2	2	0.20	900	900	900	500	900	900	200	1800
Taxa de utilização						0.985	0.987	1.000	1.000	1.000	1.000	1.000	0.128
Quantidade de caixas						93.60	93.10	94.80	53.70	95.00	95.20	19.90	21.30

Tabela 5-12: Gama tipo 1 e com nº pares por caixa a variar

Caixas	Média Pares	Delta Pares	Cap Cx	Cx Desl	Des Pad	Sec12 T1	Sec14 T2	Sec18 T3	Sec16 T4	Sec8 T5	Sec10 T6	Sec6 T7	Sec20 T8
250	5	10	2	2	0.20	5716	5716	5716	3176	5716	5716	1270	11432
Taxa de utilização						0.754	0.731	0.724	0.538	0.611	0.740	0.859	0.027
Quantidade de caixas						450.5	438.4	437.4	180.3	369.2	450.8	115.1	31.6

Tabela 5-13: Gama tipo 1 e com demasiados pedidos de caixas

Modelo G2; Período de aquecimento = 180 minutos, Tempo de análise = 480 minutos (1 dia de trabalho); 10 replicações:

Caixas	Media Pares	Delta Pares	Cap Cx	Cx Desl	Des Pad	Sec12 T1	Sec14 T2	Sec18 T3	Sec16 T4	Sec8 T5	Sec10 T6	Sec6 T7	Sec20 T8
250	5	10	2	2	0.20	1000	1000	1000	500	2500	1000	200	2500
Taxa de utilização						0.986	0.977	1.000	1.000	0.990	0.999	1.000	0.000
Quantidade de caixas						102.5	102.9	107.6	053.4	260.2	105.1	021.3	0.000

Tabela 5-14: Gama do tipo 2 e com pedidos de caixas previstos

Caixas	Media Pares	Delta Pares	Cap Cx	Cx Desl	Des Pad	Sec12 T1	Sec14 T2	Sec18 T3	Sec16 T4	Sec8 T5	Sec10 T6	Sec6 T7	Sec20 T8
250	5	10	2	2	0.20	2000	2000	2000	1000	5000	2000	400	2500
Taxa de utilização						0.979	0.983	1.000	1.000	0.912	0.977	1.000	0.000
Quantidade de caixas						205.0	208.3	214.1	106.0	478.3	207.5	042.3	0.000

Tabela 5-15: Gama tipo 2 e com o dobro de pedidos previstos

O utilizador tem a possibilidade de analisar mais em pormenor as variáveis de saída pretendidas, por exemplo ver Figura 5-3, para a secção 8 na torre 7 e para os pares pretendidos produzir, PP=2500; PP=5000 e PP=5752 a taxa de utilização varia, diminuindo para maiores solicitações de pedidos de caixas para a secção, como era de esperar.

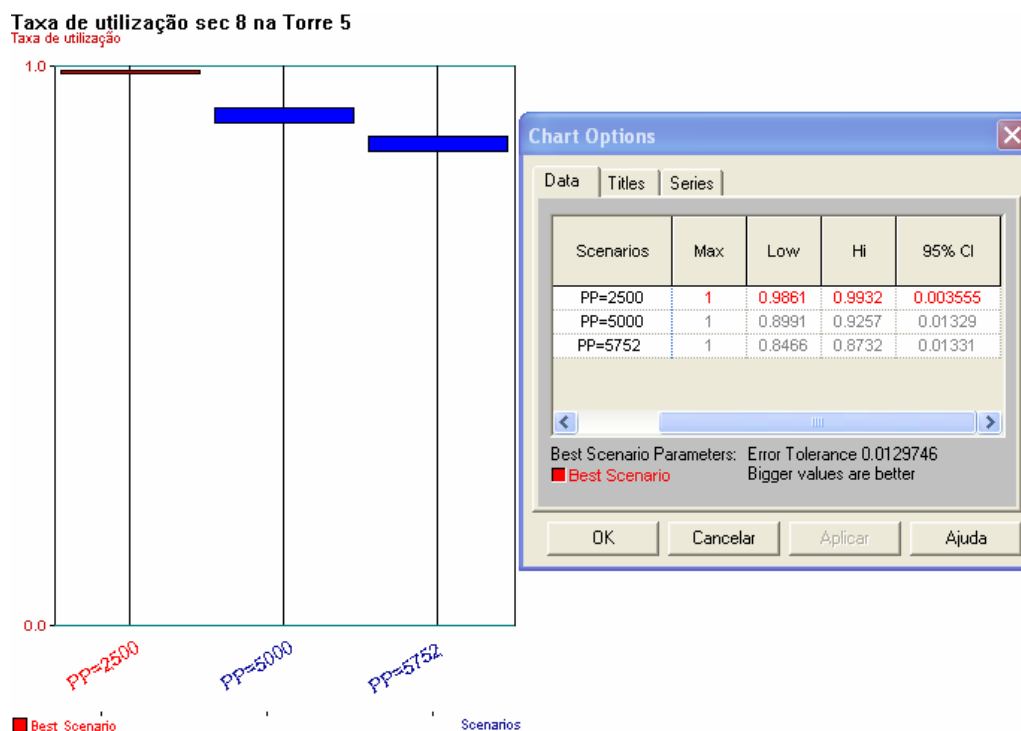


Figura 5-3: Exemplo de taxa de utilização na secção 8

No exemplo da Figura 5-3 verifica-se que a melhor estimativa da média está no cenário dos PP=2500, o meio comprimento do intervalo de confiança (half-width) é igual 0.00355, permitindo obter um melhor rigor ou confiança das estimativas encontradas.

6 Conclusões e desenvolvimentos futuros

6.1 Conclusões gerais

O trabalho descrito nesta dissertação consistiu na concepção e desenvolvimento de um modelo de simulação de um sistema de logística interna instalado numa empresa de calçado. O sistema de logística interna é composto por um sistema de movimentação e armazenamento de caixas com produto interno da empresa (pares de sapatos). A construção do modelo envolveu as seguintes fases:

- especificação do modelo conceptual de um sistema de movimentação e armazenamento de produto em curso numa empresa de calçado;
- desenvolvimento do modelo sobre a ferramenta de simulação Arena.

A melhoria do sistema de movimentação e armazenamento assume um papel importante para a melhoria dos níveis de produtividade nas empresas. O sistema de movimentação e armazenamento afecta a produtividade global na medida em que este é directamente dependente da sua eficiência e eficácia.

A visualização dinâmica do sistema modelado foi efectuada através de um modelo de animação. Este modelo foi definido através de objectos dinâmicos de animação, que permitem captar os aspectos relevantes da operação do sistema, por exemplo a ocupação dos armazéns, filas de espera, o deslocamento das caixas sobre os tapetes com cruzamentos etc. O modelo desenvolvido permite através da animação dar uma melhor percepção do que acontece ao longo do tempo da simulação. O utilizador tem ainda a possibilidade, de uma forma interactiva modificar as variáveis independentes durante a simulação e aperceber-se através da animação do estado do sistema.

O modelo de simulação desenvolvido e descrito nesta dissertação, apresenta ao utilizador a possibilidade de testar o desempenho do sistema de movimentação e armazenamento instalado na empresa de calçado para um conjunto alargado de cenários, tais como, o que acontece se aumentarmos ou diminuirmos: o número de pares médio por caixa, o número de pares produzidos em cada secção produtiva da empresa, o número de caixas em movimentação em simultâneo na direcção das

secções, a taxa de ocupação de cada armazém e a capacidade de caixas em cada secção.

Apesar de a empresa nunca ter utilizado o sistema para a capacidade prevista aquando da sua instalação, foi possível verificar através da simulação que o sistema instalado responde bastante bem às solicitações previstas. Através do ensaio de vários cenários verificou-se que o sistema instalado está sobredimensionado, como ilustrado na Tabela 5-15, para o dobro das solicitações previstas e para um conjunto de gamas operatórias específicas.

6.2 Perspectivas de desenvolvimento futuro

O sector industrial nos países desenvolvidos tem verificado um declínio em muitos sectores, incluindo o sector de calçado, principalmente devido a processos de trabalho intensivo e custos de mão-de-obra elevado. É de vital importância a optimização dos processos e em particular a redução de custos, tempos de produção e transporte entre as várias secções de produção nas empresas industriais.

Tendo sido já alcançados ganhos significativos de produtividade em diversas áreas como os processos produtivos, processos de planeamento e controlo da produção, engenharia de produto. Na área da logística interna existem ainda ganhos significativos por alcançar. Contudo, dada a diversidade de ‘layouts’ produtivos, dimensão física das instalações fabris, etc. Estes sistemas são em grande parte desenvolvidos “à medida” de cada situação, tornando o custo de concepção e desenvolvimento fora do alcance de muitas empresas, incluindo as PME’s.

O desenvolvimento de uma biblioteca de componentes de simulação de sistemas de logística, ou por outras palavras, o desenvolvimento de uma biblioteca de simulação para a modelação dinâmica das características do sistema de movimentação e armazenamento, no apoio ao projecto e desenvolvimento de soluções de automação da logística interna de empresas industriais.

Assim deste modo um dos aspectos que se pretende desenvolver futuramente é de reduzir significativamente o custo de desenvolvimento destes sistemas de logística interna, tornando-os comportáveis para um número elevado de empresas e potenciando o desenvolvimento das empresas que desenvolvem este tipo de sistemas (instaladores).

Nas empresas industriais, existe uma enorme variedade de ‘layouts’, no que respeita à movimentação e ao armazenamento de contentores. O ‘layout’ é definido em função:

- Da disposição dos tapetes ao longo da planta fabril. Apesar dos tapetes transportadores e armazéns automáticos estarem colocados a um nível elevado, os tapetes normalmente estão colocados ao longo dos corredores de circulação no interior da planta fabril.

- Da quantidade de contentores necessários a circularem em simultâneo, ou seja, a quantidade de produto em curso em cada contentor e o tipo de produtos. Deste modo poderá ser necessário a existência de um ou mais tapetes em paralelo, unidireccionais ou bidireccionais, etc.
- Capacidade de armazenamento de stock intermédio de produto em curso de fabrico. Deste modo varia a capacidade de cada armazém e o nº de armazéns necessários.
- Quantidade e distância entre secções de produção e entre secções e armazéns. Deste modo haverá velocidades variadas dos tapetes e uma complexidade de cruzamentos entre tapetes também variável.

As empresas instaladoras dos sistemas de transporte e armazenamento têm necessidade de otimizar toda a movimentação de contentores com matéria-prima e produtos em curso de fabrico na planta fabril. A optimização passa então muito genericamente por uma escolha óptima do ‘layout’ adoptado para a movimentação e armazenamento, em particular por uma boa escolha na quantificação e disposição dos tapetes, torres elevatórias e armazéns, pela escolha do tipo de tapete (velocidade do tapete, na adopção de tapetes unidireccionais ou bidireccionais, etc).

As variáveis atrás referidas tais como a escolha do tipo de tapete utilizado (unidireccional ou bidireccional), cadências do sistema, variedade de ‘layouts’, quantidade e tipo de armazéns utilizados e variedade de velocidades dos tapetes utilizados, são algumas das variáveis a estudar e analisar antes de se tomar uma decisão sobre o dimensionamento do sistema a utilizar.

Existindo uma necessidade de previamente realizar-se um estudo sobre o sistema de movimentação, a simulação aqui, tem um papel muito importante, porque permite a análise de diversas alterações no cenário virtual, sem o custo e o risco de actuar no cenário real.

Assim prevêem-se os seguintes desenvolvimentos futuros:

- O “desenvolvimento de uma biblioteca de componentes de simulação de sistemas de logística interna que permita realizar uma análise de desempenho de uma nova instalação, com um esforço reduzido”.
- A possibilidade de o utilizador da simulação poder acrescentar e retirar as gamas operatórias que se pretende incluir na simulação, para poder testar e visualizar um conjunto mais alargado de cenários. É importante, sempre que se acrescenta um modelo novo na produção ter a oportunidade de inclui-lo também na simulação e verificar o desempenho do sistema. O mesmo acontece quando acaba de ser produzido um modelo. A inclusão das gamas poderá ser realizada acrescentando uma interface gráfica, ou utilizando ficheiros de configuração a serem importados pelo sistema de simulação.

7 Referências e Bibliografia

ACT Solutions,[em linha]. [Consult. Junho 2007]. Disponível em WWW:<URL: <http://www.actsolutions.it/software/arena-simulation-rockwell.htm>>

Almeida, Marcelo, J. S. C., (1999) ATMLib – Uma Biblioteca de Classes para a Construção de Simuladores de Redes ATM: Proposta e Implementação, Dissertação de Mestrado, CCT, Universidade Federal da Paraíba, Campina Grande.

Araújo, L. C., **Souza** A. D.C., **Lima** R. Z., (2005), Manual do Arena 9.0, Avaliação de Desempenho de Sistemas

Ayres, Donald Frederick, (2000), Simulação Aplicada a Organização & Métodos no Processo de Manufatura, Monografia especialização em MBA, Universidade de Taubaté.

Banks, J., (1998), Handbook of Simulation, Principles, Methodology, Advances, Applications, and Practice, John Wiley & Sons, Inc., co-published by Engineering & Management Press.

Banks, J., (2000), Introduction to Simulation, Proceedings of the 2000 Winter Simulation Conference, J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds. pp 9-16.

Banks, J., **Carson** J.S., B.L., **Nelson**, and Nicol, D.M. (2000). Discrete event system simulation, Third Edition, Prentice Hall International Series in Industrial and Systems Engineering, Prentice Hall.

Barton, Russell R. (2003), Panel: Simulation – Past, Present and Future, Proceedings of the 2003 Winter Simulation Conference, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds. pp 2044-2050

Berends, P.; Romme, (1999) G. Simulation as a research tool in management studies. European Management Journal, v.17, n.6, pp. 576-583.

Bressan, Graça, Modelagem e Simulação de Sistemas de Computacionais, [Consult. Julho 2007]. Disponível em WWW:<URL: www.larc.usp.br/conteudo/universo/pcs012/modsim03.pdf >

Brito, António E. S. Carvalho, **Feliz**, J. Manuel Teixeira. (2001), "Simulação por Computador", Publindústria Editor.

Carson, J., **Brunner**, D. (2000), Simulation in the future: a panel discussion, Proceedings of the 2000 Winter Simulation Conference.

Carson II, J. S. (2005), Introduction to Modeling and Simulation, Proceedings of the 2005 Winter Simulation Conference, M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds. pp 16-23.

Costa, M. A. Bueno, (2002), Simulação de Aplicada à Engenharia de Produção, Universidade Federal de São Carlos.

Freitas Filho, P.J. (2001) Introdução à modelagem e simulação de Sistemas – com Aplicações em Arena. Florianópolis: Visual Books.

Guimarães, R.C. e **Cabral**, J.A.S. (1997), Estatística, McGraw-Hill

Herper, Henry, et al.(2003), Teaching The Classics Of Simulation To Beginners (Panel) Proceedings of the 2003 Winter Simulation Conference, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds. pp 1941-1951

Hlupic, Vlatka. (2000), Simulation Software: An Operational Research Society Survey of Academic and Industrial Users, Proceedings of the 2000 Winter Simulation Conference, J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds. pp1676-1683

Howell, Fred; McNab, Ross. (1997), Simjava: A Discrete Event Simulation Package for Java with Applications in Computer Systems Modelling; Department of Computer Science –The University of Edinburgh, Scotland – UK.

Kelton, W.D., **Sadowski**, Randall P. e **Sadowski**, Deborah A. (2002), “Simulation With Arena”, Second Edition, WCB McGraw-Hill.

Law, A. M. e **Kelton**, W. D. (1991), Simulation Modeling & Analysis, Second edition, McGraw-Hill, International Editions, Industrial Engineering Series.

Law, A.M. e **McComas**, M.G. (1991), Secrets of Successful Simulation Studies, Proceedings of the 1991 Winter Simulation Conference, Nelson, B.L., Kelton, W.D. and Clarck, G.M., eds. pp. 21-27.

Lula, Juliana C. L de A., (2001), Especificação de Componentes para um Ambiente de Simulação de Redes TCP/IP, Projeto de Dissertação de Mestrado, Pós-Graduação em Informática, UFPB.

Macdougall, M. H.,(1975), "System Level Simulation in Digital System Design Automation: Languages, Simulation & Data Base", Computer Science Press.

Motto, Working Group, Modular Design and Integration of Simulation Tools, [Consult. Julho 2007]. Disponível em WWW:<URL: <http://www.sim-serv.com/wg1.php#>>

Nikoukaran, Jalal, et al (1998), Criteria for Simulation Software Evaluation, Proceedings of the 1998 Winter Simulation Conference, D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, ed. pp 399-406

OR/MS Today, Simulation Software Survey ,[em linha]. [Consult. Junho 2007]. Disponível em WWW:<URL: <http://www.lionhrtpub.com/orms/surveys/Simulation/Simulation.html>>

Pedgen, C.D., **Shannon**, R.F. e **Sadowski**, R.P. (1995), Introduction to Simulation Using SIMAN, Second Edition, McGraw-Hill, Inc

Pegden, C.D. et al, (1990), Introduction to Simulation Using SIMAN, McGraw-Hill

Poorte, J.K., Johson, M.E., (1989) "A Hierarcal Aproach to Computer Animation in the Simulation Models". Simulation.

Paiva, A. F. Oliveira, (2005), Geração Automática de Modelos de Simulação de uma Linha de Produção na Indústria Têxtil, Dissertação de Mestrado, Universidade do Minho.

Prado, Darci Santos. (2004), Usando o Arena em Simulação, Segunda Edição, Terceiro Volume da Série Pesquisa Operacional, INDG Tecnologia e Serviços LTDA.

Ramos, Ana L. F. Andrade (2001), Concepção e Desenvolvimento de uma Biblioteca de Simulação para a Avaliação Dinâmica de Sistemas de Produção Celulares, Dissertação de Mestrado, Faculdade de Ciências e Tecnologias da Universidade de Coimbra.

Roberts, Chell, Dessouky, Yasser. (1998), Na Overview of Object-Oriented Simulation, Simulation, Number 70.

Saydam , T., (1995) "Process Oriented Languages", Simuleter, vol 12 (2).

Sanchez, S.M. (1999), ABC's of OutPut Analysis, Proceedings of the 1999 Winter Simulation Conference, Farrington, P.A., Nembhard, H.B., Sturroch, D.T. and Evans, G. W., eds. pp. 24-32.

Sargent, R.G. (2000), Verification, Validation, And Accreditation of Simulation Models, Proceedings of the 2000 Winter Simulation Conference, Joines, J.A., Barton, RR., Kang, K. and Fishwick, P.A., eds. pp. 50-59.

Shannon, R.E. (1998), Introduction to the Art and Science of Simulation, Proceedings of the 1998 Winter Simulation Conference, Medeiros, D.J., Watson, E.F., Carson, J.S. and Manivannan, eds. pp. 7-14.

Silva, J. E. A. Ramos (2006), Desenvolvimento de um modelo de simulação para auxiliar o gerenciamento de sistemas de corte, carregamento e transporte de cana-de-açúcar, Dissertação de mestrado, Universidade Federal de São Carlos.

Silva, Luís César, Simulação de Processos, [Consult. Janeiro 2008]. Disponível em WWW:<URL: <http://www.agais.com/simula.htm#DEB>>

Souto, Leijdane Matos (2005), Especificação e Implementação de componentes para Modelar Redes Locais sem Fio Ad Hoc Padrão IEEE 802.11, Dissertação de Mestrado, Universidade Federal de Campina Grande.

Robinson, S. (2005), Discrete-event simulation: from the pioneers to the present, what next?, Journal of the Operational Research Society (2005) 56, pp 619–629.

Rodrigues, A. Guimarães, (1996), "Simulação", Sebenta, Universidade do Minho.

Saydam, T. (1995), "Process Oriented Languages", Simuleter, vol 12 (2), Abril.

Seppanen, Marvin S. (2000), Developing Industrial Strength Simulation Models Using Visual Basic For Applications (VBA), Proceedings of the 2000 Winter Simulation Conference, J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds. pp 77-82.

Soares, Luiz F. Gomes. (1992) Modelagem e Simulação Discreta de Sistemas, VII Escola de Computação, São Paulo.

Software, Rockwell (2005), Arena Basic User's Guide, ARENAB-UM001-EN-P

Software, Rockwell (2005), Arena Standard User's Guide, ARENAB-UM001-EN-P

Software, Rockwell (2005), Variables Guide, ARENAB-UM001-EN-P

Swain, J. J. (2003), Simulation Reloaded: Sixth Biennial Survey of Discrete-Event Simulation Software, ORMS Today, Vol. 30, No. 4.

Swets, Roderick J., Drake, Glenn R. (2001), The Arena Product Family: Enterprise Modeling Solutions, Proceedings of the 2001 Winter Simulation Conference, B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds. pp 201-218.

Takus, David A., Profozich, David M. (1997), Arena Software Tutorial, Proceedings of the 1997 Winter Simulation Conference, S. Andraddttir, K. J. Healy, D. H. Withers, and B. L. Nelson

Taylor, SJE, Robinson, S. (2006), So where to next? A survey of the future for discrete-event simulation, Journal of Simulation (2006)

Tewoldeberhan, Tamrat W. et al. (2002), An Evaluation And Selection Methodology For Discrete-Event Simulation Software, Proceedings of the 2002 Winter Simulation Conference, E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds. pp 67-75.

Treadwell, Mark A., **Herrmann**, Jeffrey. (2005), A kanban an module for simulation pull production in Arena, Proceedings of the 2005 Winter Simulation Conference, M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds. pp1413-1417.

Wagner, Marcus Vinícius da Silva, (2000), Especificação de Componentes para a Simulação de Redes TCP/IP, Dissertação de Mestrado, UFPB.

Yücel Necati Deniz. (2005), Simulation of a Flexible Manufacturing System: a Pilot Implementation, Master of Science, Graduate School of Natural and Applied Sciences, Middle East Technical University

ANEXOS

A Exemplo de código

Este é um exemplo do código em VBA retirado da função “EscreveListaPedidosArmazens”

```
ArrayTorreSeccoes = TorreSeccoes(TorrePede) ' verifica quais as secções desta torre
' ver para todas as secções desta torre
For i = 0 To UBound(ArrayTorreSeccoes)
    ExisteCxArmazem = False
    If ArrayTorreSeccoes(i) <> 0 Then 'Existem secções nesta torre

        '***** ver quantidade de caixas permitido*****
        ' verificar se pode vir caixa para esta torre
        ' 1ª ver as caixas que estão na secção
        For j = 0 To UBound(ConfSeccao)
            If ConfSeccao(j, 0) = ArrayTorreSeccoes(i) Then
                If ConfSeccao(j, 4) > ConfSeccao(j, 6) Then ' verifica se o maximo permitido nesta secção é maior
                    CaixasPode1 = True
                    Exit For
                End If
            End If
        Next

        ' 2ª Verifica as caixas que estão a ir para a torre destino se maior que o máximo e as caixas no buffer da
        For k = 0 To UBound(ConfTorres)
            If ConfTorres(k, 0) = TorrePede Then
                ' verifica a quantidade de Caixas a deslocarem-se para na logistica para esta torre
                QCaixasLog = QuantasCaixasParaEsteDestino(TorrePede)
                ' If ArrayCaixasDeslTorres(TorrePede, 2) < ConfTorres(k, 3) Then ' verifica se estão a deslocar
                If QCaixasLog < ConfTorres(k, 3) Then ' verifica se estão a deslocar mais que o permitido
                    CaixasPode2 = True
                End If
                If g_SIMAN.QueueNumberOfEntities(g_SIMAN.SymbolNumber(NomeQueueEntrada)) < ConfTorres(k, 1) Then
                    CaixasPode3 = True
                End If

                If CaixasPode2 = True Or CaixasPode3 = True Then
                    Exit For
                End If
            End If
        Next

        ' verificar se estão os 3 verdadeiros
        If CaixasPode1 = True And CaixasPode2 = True And CaixasPode3 = True Then
            QCxOk = True
        End If
        '***** FIM ver quantidade de caixas permitido*****
        '***** Ver se existe alguma caixa nos armazens para esta secção*****

        Select Case TorrePede ' Evaluate Number.

        Case 1 ' Torre1

            'verifica no armazem 103
            If QCxOk = True And ExisteCxArmazem = False And TorreActiva1 = True Then ' vai ver se existe caixa
```

Figura A-1: Exemplo de Código VBA